



DFL - 800/1600/2500

User's Guide

< Version: 1.0 >

CONTENTS

I	Preface	xvi
	Document Version	xvii
	Disclaimer	xvii
	About this Document	xvii
	Typographical Conventions	xviii
II	Product Overview	2
1	Capabilities	3
	1.1 Product Highlights	3
III	Introduction to Networking	6
2	The OSI model	7
3	Firewall Principles	9
	3.1 The Role of the Firewall	9
	3.1.1 What is a Firewall?	9
	3.1.2 How does a Firewall work?	9
	3.2 What does a Firewall NOT protect against?	10
	3.2.1 Attacks on Insecure pre-installed Components	11
	3.2.2 Inexperienced Users on protected Networks	11
	3.2.3 Data-Driven Network Attacks	11
	3.2.4 Internal Attacks	13
	3.2.5 Modems and VPN Connection	13
	3.2.6 Holes between DMZs and Internal Networks	14

IV	Administration	18
4	Configuration Platform	19
4.1	Configuring Via WebUI	19
4.1.1	Overview	19
4.1.2	Interface Layout	19
4.1.3	Configuration Operations	22
4.2	Monitoring Via CLI	23
5	Logging	25
5.1	Overview	25
5.1.1	Importance & Capability	25
5.1.2	Events	26
5.2	Log Receivers	28
5.2.1	Syslog Receiver	28
5.2.2	Memory Log Receiver	29
5.2.3	SMTP Event Receiver	29
6	Maintenance	31
6.1	Firmware Upgrades	31
6.2	Reset To Factory Defaults	32
6.3	Backup Configuration	34
7	Advanced Settings	35
7.1	Overview	35
V	Fundamentals	38
8	Logical Objects	39
8.1	Address Book	39
8.1.1	IP address	39
8.1.2	Ethernet address	41
8.2	Services	41
8.2.1	Service Types	42
8.2.2	Error Report & Connection Protection	46
8.3	Schedules	48
8.4	X.509 Certificates	49
8.4.1	Introduction to Certificates	49
8.4.2	X.509 Certificates in D-Link Firewall	51

9	Interfaces	53
9.1	Ethernet	53
9.1.1	Ethernet Interfaces	53
9.1.2	Ethernet Interfaces in D-Link Firewalls	54
9.2	Virtual LAN (VLAN)	56
9.2.1	VLAN Infrastructure	56
9.2.2	802.1Q VLAN Standard	57
9.2.3	VLAN Implementation	58
9.2.4	Using Virtual LANs to Expand Firewall Interfaces	59
9.3	DHCP	60
9.3.1	DHCP Client	60
9.4	PPPoE	61
9.4.1	PPP	62
9.4.2	PPPoE Client Configuration	62
9.5	Interface Groups	65
9.6	ARP	66
9.6.1	ARP Table	66
10	Routing	69
10.1	Overview	69
10.2	Routing Hierarchy	70
10.3	Routing Algorithms	71
10.3.1	Static Routing	71
10.3.2	Dynamic Routing	72
10.3.3	OSPF	74
10.4	Route Failover	77
10.4.1	<i>Scenario</i> : Route Failover Configuration	78
10.5	Dynamic Routing Implementation	81
10.5.1	OSPF Process	81
10.5.2	Dynamic Routing Policy	81
10.5.3	<i>Scenarios</i> : Dynamic Routing Configuration	82
10.6	<i>Scenario</i> : Static Routing Configuration	87
10.7	Policy Based Routing(PBR)	88
10.7.1	Overview	88
10.7.2	Policy-based Routing Tables	89
10.7.3	Policy-based Routing Policy	89
10.7.4	PBR Execution	89
10.7.5	<i>Scenario</i> : PBR Configuration	91
10.8	Proxy ARP	94

11 Date & Time	95
11.1 Setting the Date and Time	96
11.1.1 Current Date and Time	96
11.1.2 Time Zone	96
11.1.3 Daylight Saving Time(DST)	97
11.2 Time Synchronization	98
11.2.1 Time Synchronization Protocols	98
11.2.2 Timeservers	98
11.2.3 Maximum Adjustment	99
11.2.4 Synchronization Interval	99
12 DNS	101
13 Log Settings	103
13.1 Implementation	103
13.1.1 Defining Syslog Receiver	103
13.1.2 Enabling Logging	104
VI Security Polices	108
14 IP Rules	109
14.1 Overview	109
14.1.1 Fields	110
14.1.2 Action types	111
14.2 Address Translation	112
14.2.1 Overview	112
14.2.2 NAT	112
14.2.3 Address translation in D-Link Firewall	114
14.3 <i>Scenarios</i> : IP Rules Configuration	116
15 Access (Anti-spoofing)	123
15.1 Overview	123
15.1.1 IP Spoofing	123
15.1.2 Anti-spoofing	124
15.2 Access Rule	124
15.2.1 Function	124
15.2.2 Settings	124
15.3 <i>Scenario</i> : Setting up Access Rule	126

16 DMZ & Port Forwarding	127
16.1 General	127
16.1.1 Concepts	127
16.1.2 DMZ Planning	129
16.1.3 Benefits	130
17 User Authentication	131
17.1 Authentication Overview	131
17.1.1 Authentication Methods	131
17.1.2 Password Criterion	132
17.1.3 User Types	133
17.2 Authentication Components	134
17.2.1 Local User Database(UserDB)	134
17.2.2 External Authentication Server	134
17.2.3 Authentication Agents	135
17.2.4 Authentication Rules	136
17.3 Authentication Process	137
17.4 <i>Scenarios: User Authentication Configuration</i>	137
VII Content Inspection	146
18 Application Layer Gateway (ALG)	147
18.1 Overview	147
18.2 FTP	148
18.2.1 FTP Connections	148
18.2.2 <i>Scenarios: FTP ALG Configuration</i>	150
18.3 HTTP	155
18.3.1 Components & Security Issues	155
18.3.2 Solution	156
18.4 H.323	158
18.4.1 H.323 Standard Overview	158
18.4.2 H.323 Components	158
18.4.3 H.323 Protocols	159
18.4.4 H.323 ALG Overview	160
18.4.5 <i>Scenarios: H.323 ALG Configuration</i>	161
19 Intrusion Detection System (IDS)	181
19.1 Overview	181
19.1.1 Intrusion Detection Rules	182
19.1.2 Pattern Matching	182

19.1.3 Action	182
19.2 Chain of Events	183
19.2.1 Scenario 1	183
19.2.2 Scenario 2	184
19.3 Signature Groups	186
19.4 Automatic Update of Signature Database	186
19.5 SMTP Log Receiver for IDS Events	187
19.6 <i>Scenario: Setting up IDS</i>	189
VIII Virtual Private Network (VPN)	192
20 VPN Basics	193
20.1 Introduction to VPN	193
20.1.1 VPNs vs Fixed Connections	193
20.2 Introduction to Cryptography	195
20.2.1 Encryption	195
20.2.2 Authentication & Integrity	198
20.3 Why VPN in Firewalls	200
20.3.1 VPN Deployment	201
21 VPN Planning	207
21.1 VPN Design Considerations	207
21.1.1 End Point Security	208
21.1.2 Key Distribution	210
22 VPN Protocols & Tunnels	213
22.1 IPsec	213
22.1.1 IPsec protocols	214
22.1.2 IPsec Modes	214
22.1.3 IKE	215
22.1.4 IKE Integrity & Authentication	219
22.1.5 <i>Scenarios: IPSec Configuration</i>	223
22.2 PPTP/ L2TP	228
22.2.1 PPTP	228
22.2.2 L2TP	234
22.3 SSL/TLS (HTTPS)	243

IX	Traffic Management	246
23	Traffic Shaping	247
23.1	Overview	247
23.1.1	Functions	248
23.1.2	Features	249
23.2	Pipes	249
23.2.1	Precedences and Guarantees	250
23.2.2	Grouping Users of a Pipe	252
23.2.3	Dynamic Bandwidth Balancing	253
23.3	Pipe Rules	253
23.4	<i>Scenarios</i> : Setting up Traffic Shaping	253
24	Server Load Balancing (SLB)	261
24.1	Overview	261
24.1.1	The SLB Module	261
24.1.2	SLB Features	262
24.1.3	Benefits	263
24.2	SLB Implementation	264
24.2.1	Distribution Modes	264
24.2.2	Distribution Algorithms	264
24.2.3	Server Health Checks	265
24.2.4	Packets Flow by SAT	266
24.3	<i>Scenario</i> : Enabling SLB	266
X	Misc. Features	270
25	Miscellaneous Clients	271
25.1	Overview	271
25.2	Dynamic DNS	271
25.3	Automatic Client Login	272
25.4	HTTP Poster	273
25.4.1	URL Format	273
26	DHCP Server & Relay	275
26.1	DHCP Server	275
26.2	DHCP Relay	277

XI	Transparent Mode	280
27	Transparent Mode	281
27.1	Overview	281
27.2	Transparent Mode Implementation in D-Link Firewalls	282
27.3	<i>Scenarios: Enabling Transparent Mode</i>	284
XII	Zone Defense	292
28	Zone Defense	293
28.1	Overview	293
28.2	Zone Defense Switches	293
28.2.1	SNMP	294
28.3	Threshold Rules	295
28.4	Manual Blocking & Exclude Lists	295
28.5	Limitations	296
28.6	<i>Scenario: Setting Up Zone Defense</i>	296
XIII	High Availability	300
29	High Availability	301
29.1	High Availability Basics	301
29.1.1	What High Availability will do for you	301
29.1.2	What High Availability will NOT do for you	302
29.1.3	Example High Availability setup	303
29.2	How Rapid Failover is Accomplished	303
29.2.1	The shared IP address and the failover mechanism	304
29.2.2	Cluster heartbeats	305
29.2.3	The synchronization interface	306
29.3	Setting up a High Availability Cluster	306
29.3.1	Planning the High Availability cluster	307
29.3.2	Creating a High Availability cluster	307
29.4	Things to Keep in Mind	309
29.4.1	Statistics and Logging Issues	309
29.4.2	Configuration Issues	310
XIV	Appendix	312
A	Console Commands Reference	315

List of Commands	315
About	315
Access	316
ARP	316
ARPSnoop	317
Buffers	317
Certcache	318
CfgLog	319
Connections	319
Cpuid	320
DHCP	320
DHCPRelay	321
DHCPServer	321
DynRoute	321
FragS	322
HA	322
HTTPPoster	322
Ifacegroups	323
IfStat	323
Ikesnoop	324
Ipseckeealive	325
IPSecTunnels	325
IPSecstats	325
Killsa	326
License	326
Lockdown	327
Loghosts	327
Memory	327
Netcon	327
Netobjects	328
OSPF	328
Ping	329
Pipes	329
Proplists	330
ReConfigure	330
Remotes	331
Routes	331
Rules	332
Scrsave	332
Services	333
Shutdown	333

Sysmsgs	333
Settings	333
Stats	335
Time	336
Uarules	336
Userauth	336
Userdb	337
Vlan	338
B Customer Support	341

FIGURES & TABLES

2.1	The OSI 7-Layer Model.	8
4.1	WebUI Authentication Window.	20
4.2	WebUI Main Display.	20
9.1	A VLAN Infrastructure.	57
9.1	802.1Q Standard Ethernet Frame.	58
10.1	Route Failover Scenario	78
10.2	OSPF Process Scenario	82
10.3	Static Routing Scenario	87
14.1	Dynamic NAT.	114
14.1	SAT Example.	119
16.1	A Web Server in DMZ	128
18.1	FTP ALG Scenario 1	150
18.2	FTP ALG Scenario 2	153
18.3	H.323 Scenario 1.	162
18.4	H.323 Scenario 2.	166
18.5	H.323 Scenario 3.	169
18.6	H.323 Scenario 4.	172
18.7	H.323 Scenario 5.	174
19.1	IDS Chain of Events Scenario 1	183
19.2	IDS Chain of Events Scenario 2	185
19.3	Signature Database Update	187

19.4 An IDS Scenario	189
20.1 VPN Deployment Scenario 1	201
20.2 VPN Deployment Scenario 2	202
20.3 VPN Deployment Scenario 3	203
20.4 VPN Deployment Scenario 4	203
20.5 VPN Deployment Scenario 5	204
20.6 VPN Deployment Scenario 6	205
22.1 LAN-to-LAN Example Scenario.	223
22.2 IPSec Roaming Client Example Scenario.	225
22.1 PPTP Encapsulation.	228
22.2 L2TP Encapsulation.	235
23.1 IPv4 Packet Format	251
24.1 A SLB Logical View.	262
24.2 A SLB Scenario	266
27.1 Transparent Mode Scenario 1.	284
27.2 Transparent Mode Scenario 2.	287
28.1 A Zone Defense Scenario.	297
29.1 Example HA Setup.	303

LIST OF SCENARIOS

<i>Section 10.4:</i> Route Failover Configuration	78
<i>Section 10.5:</i> Dynamic Routing Configuration	82
<i>Section 10.6:</i> Static Routing Configuration	87
<i>Section 10.7:</i> PBR Configuration	91
<i>Section 14.3:</i> IP Rules Configuration	116
<i>Section 15.3:</i> Setting up Access Rule	126
<i>Section 17.4:</i> User Authentication Configuration	137
<i>Section 18.2:</i> FTP ALG Configuration	150
<i>Section 18.4:</i> H.323 ALG Configuration	161
<i>Section 19.6:</i> Setting up IDS	189
<i>Section 22.1:</i> IPSec Configuration	223
<i>Section 23.4:</i> Setting up Traffic Shaping	253
<i>Section 24.3:</i> Enabling SLB	266
<i>Section 27.3:</i> Enabling Transparent Mode	284
<i>Section 28.6:</i> Setting Up Zone Defense	296

Part I

Preface

Document Version

Version No.: 1.0

Disclaimer

Information in this user's guide is subject to change without notice.

About this Document

This User's Guide is designed to be a handy configuration manual as well as an Internetworking and security knowledge learning tool for network administrators.

The document attempts not only to present means for accomplishing certain operations of the product, but provides fundamentals on what concepts the functions are based on, how the various sections of the product actually work, and why a certain set of configurations is performed, in order to enhance the reader's understanding.

The content of this guide is logically organized to *Parts*, *Chapters*, and *Sections*, with *Scenario* analysis for every main feature, to better enable the reader to learn various functions. Following the detailed parts and chapters, supplemental information and an index of relevant terms in this guide are presented.

Typographical Conventions



Example:

Configuration steps for achieving certain function.

WebUI :

Example steps for WebUI.

■ Note ■

■ Additional information the user should be aware of. ■



Tip

Suggestions on configuration that may be taken into consideration.



Caution

Critical information the user should follow when performing certain action.



Warning

Critical information the user MUST follow to avoid potential harm.

*P*_{art} *II*

Product Overview

CHAPTER 1

Capabilities

1.1 Product Highlights

The key features of D-Link firewalls can be outlined as:

- Easy to use start-up wizard
- Web-based graphical user interface (WebUI)
- Effective and easy to maintenance
- Complete control of security policies
- Advanced application layer gateways (FTP, HTTP, H.323)
- Advanced monitoring & logging methods
- Full VLAN compliance
- Support for building VPN (IPSec, PPTP, L2TP)
- Route Failover
- Advanced routing (OSPF)
- Transparent Mode support
- Server Load Balancing
- Intrusion Detection System

- Zone Defence
- High Availability (Some models)

Details about how to make these features work can be found in specific chapters in this user's guide.

Part III

Introduction to Networking

CHAPTER 2

The OSI model

Open System Interconnection (OSI) model defines a primary framework for intercomputer communications, by categorizing different protocols for a great variety of network applications into seven smaller, more manageable layers. The model describes how data from an application in one computer can be transferred through a network medium to an application in another computer. The control of the data traffic is passed from one layer to the next, starting at the application layer in one computer, proceeding to the bottom layer, traversing over the medium to another computer and then delivering up to the top of the hierarchy. Each layer handles a certain set of protocols, so that the tasks for achieving an application can be distributed to different layers to be implemented independently.

Table 2.1 shows the definition of the 7 layers. The basic functions and common protocols involved in each layer are explained below.

Application Layer

- defines the user interface that supports applications directly.
Protocols: HTTP, FTP, DNS, SMTP, Telnet, SNMP, etc.

Presentation Layer

- translates the various applications to uniform network formats that the rest of the layers can understand.

Session Layer

- establishes, maintains and terminates sessions across the network.
Protocols: NetBIOS, RPC, etc.

7	Application Layer
6	Presentation Layer
5	Session Layer
4	Transport Layer
3	Network Layer
2	Data-Link Layer
1	Physical Layer

Table 2.1: The OSI 7-Layer Model.

Transport Layer

– controls data flow and provides error-handling. *Protocols:* TCP, UDP, etc.

Network Layer

– performs addressing and routing. *Protocols:* IP, OSPF, ICMP, IGMP, etc.

Data-Link Layer

– frames the data. *Protocols:* Ethernet, PPP, etc.

Physical Layer

– defines hardware supports.

D-Link firewalls handle network traffics and perform diverse functions for security guarantee and application support throughout the 7 layers of the OSI model.

CHAPTER 3

Firewall Principles

3.1 The Role of the Firewall

3.1.1 What is a Firewall?

When you connect your computer or your local area network to another network, e.g. the Internet, measures need to be taken to prevent intruders from gaining access to resources and material you consider confidential or sensitive. In order to achieve this, a firewall must be implemented in the network. Its task is to ensure that only approved communication is allowed to flow between networks and that unauthorized communication is blocked and logged.

3.1.2 How does a Firewall work?

The primary purpose of a firewall is to enforce a security policy stating who can communicate with whom and in what way.

The firewall accomplishes this by examining the traffic that passes through it, comparing this information to a set of rules programmed into it and making a decision based on factors such as sender address, destination address, protocol and ports. This allows you to install less secure network services on your protected networks and prevent all outsiders from ever gaining access to these services.

Most firewalls, including D-Link firewalls, ensure that network traffic

complies with current protocol definitions. This can prevent poorly implemented services on the protected servers and client software from being exposed to unexpected data, causing them to hang or crash. In short, a firewall is the network's answer to poor host security.

3.2 What does a Firewall NOT protect against?

Security means much more than just firewalls. However, in most cases, installing a firewall is a necessary first step towards securing your network and computers.

This section is not specifically devoted to D-Link firewalls; instead it discusses firewalls in general. The problems described here will occur no matter which firewall you choose to install.

A common misconception is that all communication is immediately made safe and secure once it passes through a firewall. This is however not true.

Many marketing executives and sales people smile and claim that "our firewall will protect you against everything". We hope that this is just sheer ignorance on their part and not a conscious attempt to mislead potential buyers.

A firewall can only protect against that for which it was designed. Unfortunately, it is impossible to predict all the bugs other software may have. In addition, there are a large number of situations where a firewall quite simply cannot provide protection since not all communication passes through it.

The following is a selection of security problems that firewalls are often unable to deal with, and in some instances we have provided solutions to combat these.

Please note that this only scratches the surface in terms of the number of existing problems.

Complete protection can only be achieved through thorough understanding of all possible weaknesses in network protocols and in the software used, and by implementing appropriate measures to compensate for these.

3.2.1 Attacks on Insecure pre-installed Components

A very common problem is the fact that operating systems and applications usually contain insecure pre-installed components. Such components include undocumented services present on computers connected to the Internet, allowing inbound external network connections. One example of this form of vulnerability is the "simplifying" components that allow direct ODBC access via HTTP in web servers.

The common feature of most of these components is that they are not intended for use on a public network, where intruders can utilize the extra functionality at hand to easily break into the system. However, modern systems are frequently supplied with such components pre-installed in order to make the system easier to use.

A good precaution to take is to review all Internet-connected systems, clients and servers, and remove all unnecessary functionality.

3.2.2 Inexperienced Users on protected Networks

No firewall in the world can protect against the damage that inexperienced users can do to a protected network.

If they "assist" an intruder in one way or another, e.g. by opening an unrecognized program sent to them by email such as "merryxmas2001.exe", they can do more damage than all the bugs in applications and operating systems put together.

All attempts to secure the networks of an organization should be preceded by a thorough investigation of what should and should not be permitted. The result of this should be a security policy that applies to all parts of the organization, from management down. In order for such a policy to work, all users must be made aware of this policy and why it must be enforced.

3.2.3 Data-Driven Network Attacks

Normally, a firewall will only protect a system against data-driven attacks in exceptional circumstances. Such attacks include:

- HTML pages containing javascript or Java that attack the network "from the inside" when the page is viewed in a browser or e-mail program. The only possible protection against this sort of attack,

apart from better written software, is to disable such services or limiting surfing to less sensitive computers.

- HTML pages that link in the contents of local files when they are opened without scripts. Such pages can, often with the help of unsuspecting local users who are lured into "helping" the page by clicking on a button, send the linked file onwards to an unknown Internet server.
- Documents sent by email that contain hostile scripts which are activated once the document is opened. Possible ways to protect your system against this form of attack include avoiding using browser-based email software or disabling scripting and introducing mail gateways that can block scripts and other executable code.
- Buffer overruns, which firewalls only rarely provide protection against. Buffer overruns can occur in any application, with a net result of intruders being able to coax protected computers into executing any command. Here, the only solution is to ensure that only well-written applications, which are specifically designed to be immune to this form of attack are installed and used. Unfortunately, most current software is not written with this problem in mind. At the time of writing, we are of the opinion that this poses the greatest technical threat of all forms of network-based attack, as almost all software is susceptible to buffer overruns.
- Viruses and Trojan horses. A firewall can of course be connected to virus scanners, mail gateways and other similar devices in order to increase security, but it should be noted that the fundamental functionality of a firewall does not normally provide such protection.
- Even if the firewall is connected to a virus scanner, it is possible that attacking viruses could be so well hidden that the scanner would be unable to detect them. In addition, a virus scanner can only detect viruses it recognizes. If somebody designs a virus specifically for attacking your systems or those of a small group of people, or if the trojan or virus in question has not been in circulation long enough for it to become well known, the virus scanner will not recognize it.

At present, the most common targets for data-driven attacks are:

- Public servers such as mail servers, DNS servers and web servers. Web servers are clearly over-represented in this category due to their enormous complexity.

- Customized scripts on web servers. It is now very easy to extend the functionality of your web server by writing small, customized programs to handle a multitude of tasks. However, insufficient awareness of potential problems can lead you, more often than not, to make small, difficult to detect mistakes that will enable an intruder to gain access to your system.
- Web browsers. Automation of processes and simplifying operations for the benefit of users creates increased internal complexity and thereby increased risks of vulnerabilities.
- Desktop software, primarily that which to great extent support scripting languages, for the same reason as browsers. Scripting languages provide almost unlimited access to local computers and all connected network resources. As a result, intruders can cause all types of problems if they can get internal users to open documents containing malevolent scripts.

3.2.4 Internal Attacks

A firewall can only filter data that passes through it. Therefore it can't offer any protection from internal attacks on local networks, where all computers communicate directly with each other.

In addition, firewalls cannot provide protection against local users introducing harmful software to the network from a removable media, or by exporting sensitive information in the same manner.

This may seem obvious. However, most people underestimate the impact of such damage.

Although different sources provide different figures, it is clear that more than 50% of all data security problems are the results of internal attacks. Some sources put this figure as high as 80%.

3.2.5 Modems and VPN Connection

A common misconception is that modems and VPN gateways are as secure as the protected network and can be connected directly to it without protection.

Modem pools can be subject to direct attacks and, in extreme cases, telephone lines can be tapped. Switches, located at any point in the telecommunications network or in the office, can be reprogrammed without the intruder needing to be anywhere near them.

When it comes to VPN connections, it is important to remember that although the connection itself may be secure, the total level of security is only as high as the security of the tunnel endpoints.

It is becoming increasingly common for users on the move to connect directly to their company's network via VPN from their laptops. However, the laptop itself is often not protected. In other words, an intruder can gain access to the protected network through an unprotected laptop with already-opened VPN connections.

A basic precaution to take in protecting your network against modem and VPN connection attacks is to ensure that mobile computers never communicate directly with the Internet. Instead, they should always be routed through the VPN or modem connection and the company's network, regardless of whom they wish to communicate with. This way, they enjoy more or less the same level of protection as the rest of the network. For VPN connections, a competent VPN client that can block all inbound Internet traffic, aside from that which passes through the VPN connection, must be installed on each laptop.

A VPN connection or modem pool should never be regarded as a direct part of a protected network. The VPN endpoints should instead be located in a special DMZ or outside a firewall that is dedicated to this task. By doing this, you can restrict which services can be accessed via VPN and modem and therefore ensure that these services are well protected against intruders.

In instances where the firewall features an integrated VPN gateway, it is usually possible to dictate the types of communication permitted. The D-Link Firewall features just such a facility.

3.2.6 Holes between DMZs and Internal Networks

Although the advent of extranets and e-commerce has served to drive development forwards, and as more and more companies begin to make internal data available via web servers, security hazards are increasing as a

result.

It is now common practice to locate web servers in demilitarized zones, where they communicate with data sources on protected networks. In such cases, data-driven attacks pose a huge threat.

The problem with holes between DMZs and internal networks is not really a problem in itself. Rather, it is a consequence of the problems discussed above. Many people open up these holes without being aware of the problems they may cause, which is why we have chosen to highlight this problem in a separate section.

The reason for locating a web server in a DMZ is simple - the server cannot be relied upon to be completely secure. What happens if someone gains control over the server and there is an open hole through which access can be gained to data sources on the internal network? The result is that the "protected" network is open to attack from the Internet, using the web server as an intermediate.

Do not underestimate the effects of this vulnerability! In our experience, even the most inexperienced of attackers need only a few minutes to gain access to protected networks using standardized and well-known techniques, specifically developed to exploit this type of hole.

The simplest defense against this is increased segmentation of the network. By locating the data source, e.g. an SQL server, in a separate network segment and preventing it from communicating directly with the rest of the network, you can limit the damage caused by such an attack.



■ The problem here is not IP packets being routed via the servers in the DMZ, so therefore disabling "IP forwarding" would not provide any protection. The problem is that intruders can execute commands on these servers the same way that anyone at the keyboard could. ■

It should also be noted that your internal network would still be vulnerable to attack even if the channel between the DMZ and the internal network is made up of a non-routable protocol such as NetBEUI. Again, the problem is not IP packets traversing from insecure networks to the internal network.

Rather, the problem is that insecure machines can execute commands on "protected" machines.

Another form of protection worth considering is to set up a separate data source that contains limited information to which the web server has access. It should only contain information deemed sufficiently insensitive to be accessed from the web server. This process requires an automatic export of data from the internal data source to the external data source, to be executed when information needs to be updated, or at fixed times of the day. An insurmountable problem may arise when the web server needs to update the data source. The best way of tackling such a problem is to move the affected data source to a separate network segment, thereby decreasing the potential damage in the case of intrusion.

Part IV

Administration

This part covers basic aspects of D-Link firewall management and administration, including:

- [Configuration Platform](#)
- [Logging](#)
- [Maintenance](#)
- [Advanced Settings](#)

CHAPTER 4

Configuration Platform

4.1 Configuring Via WebUI

4.1.1 Overview

The D-Link firewall can be configured using a web interface. A web interface is usually a fast and efficient way to configure a firewall, that does not require the administrator to install any specific programs to configure the firewall. This will also allow the administrator to configure the firewall remotely, virtually from anywhere in the world.

4.1.2 Interface Layout

Before using the WebUI interface, the user will have to be authenticated by entering username/password in the authentication window, shown in Figure 4.1.

Once logged into the WebUI, the user will be presented a page with three distinct sections, as shown in Figure 4.2:

- Menu Bar
- Tree-view List
- Main Window



Figure 4.1: WebUI Authentication Window.

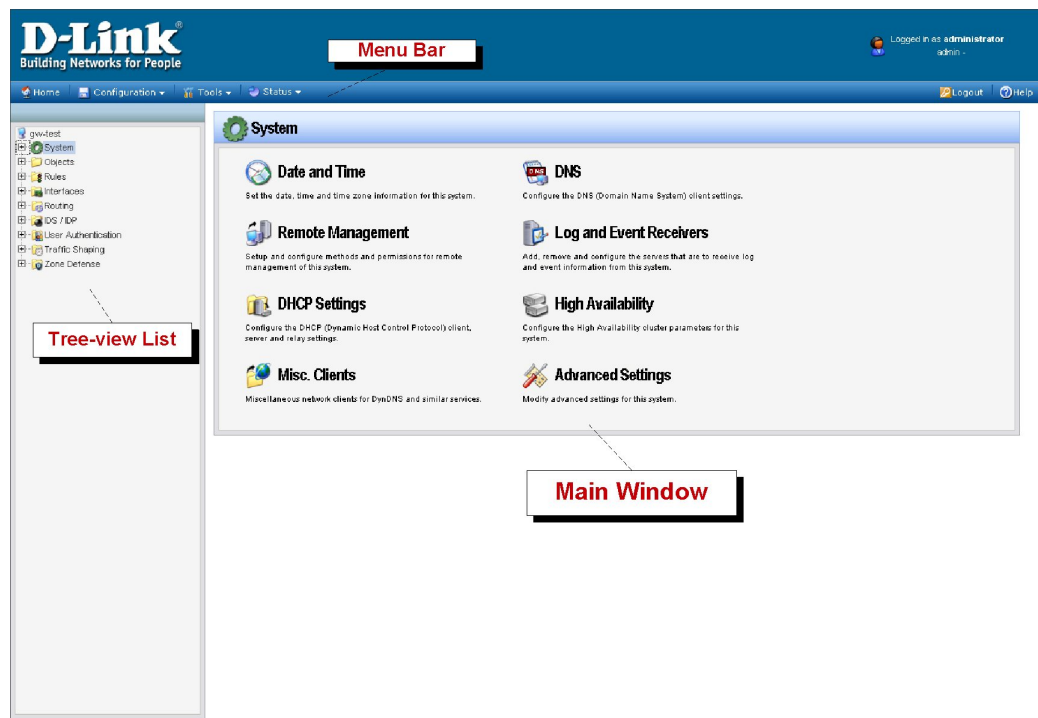


Figure 4.2: WebUI Main Display.

Menu Bar

The menu bar consists of a number of buttons with either a single option or multiple suboptions.

- Home
 - Go to start page of WebUI.
- Configuration
 - Save and Activate: Save the configuration and activate the changes.
 - Discard Changes: Discard latest changes in the configuration.
- Tools
 - Ping: Tool used to ping hosts in the network. Useful for problem solving and debugging.
 - Backup: Tool used to save and restore backups of the current configuration.
 - Reset: On this page it is possible to restart the firewall and restore it to factory default.
 - Upgrade: On this page the IDS signatures and firewall firmware can be upgraded.
- Status
 - System: Here system status is shown. CPU load, connections etc.
 - Logging: This is where the logs stored in the memory logger is displayed.
 - Connections: Displays current connections through the firewall.
 - Interfaces: Displays status for interfaces and tunnels.
 - IPSec: Displays IPSec status information.
 - Routes: Displays the current routing table.
 - DHCP Server: Displays usage information for DHCP servers.
 - IDS: Displays IDS status information.
 - SLB: Displays SLB status information.

- Zone Defense: Displays Zone Defense status information.
- Logout
Log out from the WebUI.
- Help
Read the latest version of this manual.

Tree-view List

The tree view list is a listing of the configuration sections in the firewall. The tree can be expanded to show more detailed configuration options.

Main Window

The main window displays the selected configuration section or the object to modify. Make sure to click on the OK button to save changes made to an object, or cancel to discard them, before navigating further in the WebUI.

4.1.3 Configuration Operations

When configuring the firewall, the same IP addresses, network definitions, services etc, are often used in multiple locations throughout the configuration. To simplify administration and make it easier to change IP addresses, networks etc, logical objects (see [8 Logical Objects](#)) are used throughout the firewall configuration.

When the user has configured the firewall via the WebUI, the configuration will have to be saved and activated before the new configuration will be used by the firewall. This is done via the "**Save and Activate**" menu bar option under "**Configuration**".

4.2 Monitoring Via CLI

Administrators can also monitor and troubleshoot the D-Link firewall via *Command-Line Interface (CLI)*, by employing the *Console* port on the firewall.

The serial console port is a RS-232 port that enables a connection to a PC or terminal. To access the console port, the following equipments are needed:

- A terminal or a (portable) computer with a serial port and the ability to emulate a terminal (i.e. using the **Hyper Terminal software** included in most Microsoft Windows installations). The terminal should have the following settings: **9600 baud, No parity, 8 bits and 1 stop bit**.
- A RS-232 cable with appropriate connectors.

To connect a terminal to the console port, follow these steps:

1. Set the terminal settings as described previously.
2. Connect one of the connectors of the RS-232 cable directly to the console port on the firewall hardware.
3. Connect the other end of the cable to the terminal or the serial connector of the computer running the communication software.

Through the text-based command line interface (CLI) from the console, a more in-depth analysis of various statistical aspects of the firewall can be conducted as well as advanced troubleshooting. A detailed reference of various commands that can be used in this interface is covered in [Appendix A, Console Commands Reference](#).



■ Currently, the CLI can only be used for statistics and status display. The firewall can NOT be configured via this interface. ■

CHAPTER 5

Logging

This chapter discusses principles of logging and gives a brief introduction to D-Link firewalls' logging design. For information about how to implement logging function by the firewall, please refer to [13, Log Settings](#) in the Fundamentals part.

5.1 Overview

Logging is a practice to keep track of activities that pertinent to firewall operation and the security policy the firewall is enforcing. The log file generated from logging helps administrators to observe in details of what events have occurred. D-Link firewalls provide a variety of options for logging its activities.

5.1.1 Importance & Capability

Regardless of what security policy is being implemented by the firewall, logging is critical to ensure that the implementation is running smoothly as well to keep an eye on what is going on in an network environment. It gives professionals the ability to monitor the operation of the device and assure that events in progress are expected.

Since the firewall is taking charge of all traffic that going through its interfaces from protected network to other areas and also the other way around, any misconfiguration or misuse of the functions might result in

discontinuity of services. By reviewing the output of logging, there is a good chance that the administrator will be able to figure out the problematic events, and take necessary actions to correct the problems. Once the problem is resolved, the correct content can be found in the new logging information to verify that proper changes have been done.

Logging can also be used in *Intrusion Detection System (IDS)*. The suspect traffic and attempted, failed, or successful attacks against the firewall and the network can be record, with notifications sent to alert administrators. These logging information is very useful for administrators to determine how an intrusion might have occurred and what counter-attack method can be added to improve the firewall's implementation.

As soon as log-required events are taking place, the firewall generates responses based on those events, and the responses are output into log files of one form or another to one or more log receivers.

5.1.2 Events

There are a number of different situations that will cause D-Link firewalls to generate and deliver log data. Each such occasion is referred to as an *event*.

Some events, for instance, the firewall's startup and shutdown, will always generate log entries. Others, for instance, to log if a specified rule is being matched, are configurable. The most obvious and straight-forward reason for event generating is, of course, when logging is configured in the firewall's rules, such as IP rules, User Authentication rules, Threshold rules, and so on.

Events of interest for capturing generally fall into three broad categories: *Firewall System Issues*, *Security Policy*, and *Network Connection Status*.

System Issues

This category of events logs the firewall system's status and hardware changes, for instance:

- **BUFFERS**– events regarding buffer usage.
- **TIMESYNC**– firewall time synchronization events.
- **HWM**– hardware monitor events.
- **SYSTEM**– startup & shutdown

Security Policy

Information about different actions triggered by the firewall's rules are given in this category, including:

- **ACCEPT**– packets accepted for further transmission.
- **FWD**– packets statelessly forwarded.
- **DROP**– packets disallowed.

Network Connection

Various traffic connections, routing status, and user activities record for network environment debug and monitor fall into this category. Both authorized services and rejected connections can be logged. Normally, the name of the service (or protocol name) is used as the *Flag* for the event in the log entry. The most common events within this category are listed below.

- **USAGE**
 - periodical system usage statistics, such as bandwidth, connections, and etc.
- **CONN**
 - state engine events, e.g. open/close connections.
- **NETCON**
 - administrator's remote management events.
- **IFACEMON**
 - interface monitor events.
- **DHCP/DHCPRELAY/DHCPSERVER**
 - events for DHCP client, relay, or server.
- **ARP**
 - log messages coming from the ARP engine.
- **FRAG**
 - log messages coming from the fragment handling engine.
- **OSPF/DYNROUTING**
 - information for dynamic routing.
- **RFO**
 - route fail over events.
- **PPP/PPPOE/PPTP/L2TP/GRE/IPSEC**
 - events for different tunnels.
- **USERAUTH**
 - events for user authentication.

- **HA**
 - High Availability events.
- **IDS/IDSUPDATE**
 - Intrusion Detection events and database update.
- **ZONEDEFENSE**
 - Zone Defense events.
- **SNMP**
 - allowed and disallowed SNMP accesses.
- **IP.../TCP...**
 - information concerning TCP/IP packets.

5.2 Log Receivers

A log receiver is a separate computer, known as "Syslog server", or a memory section built in the firewall to handle all the logged events generated by the firewall.

Once a new event is received, the receiver adds an entry into the log file to record the data.

5.2.1 Syslog Receiver

D-Link Firewall can send log data to syslog recipients. Syslog is a standardized protocol for sending log data to loghosts, although there is no standardized format of these log messages. The format used by D-Link Firewall is well suited for automated processing, filtering and searching.

Although the exact format of each log entry depends on how a particular syslog recipient works, most are very much alike. The way in which logs are read is also recipient dependent. Syslog daemons on UNIX servers usually log to text files, line by line.

Most syslog recipients preface each log entry with a timestamp and the IP address of the machine that sent the log data:

```
Feb 5 2000 09:45:23 gateway.ourcompany.com
```

This is followed by the text the sender has chosen to send. All log entries from D-Link Firewall are prefaced with "FW:" and a category, e.g. "DROP:".

Feb 5 2000 09:45:23 gateway.ourcompany.com FW: DROP:

Subsequent text is dependent on the event that has occurred.

In order to facilitate automated processing of all messages, D-Link Firewall writes all log data to a single line of text. All data following the initial text is presented in the format **name=value**. This enables automatic filters to easily find the values they are looking for without assuming that a specific piece of data is in a specific location in the log entry.

In a D-Link firewall, up to 8 Syslog receivers can be configured, and they can be grouped into one or more receiver groups.

Compared to the *Memory Log Receiver* which is introduced next, Syslog receivers can be used for safer and long-term storage of logged events. These log servers provide centralized management of log files, and backup of the files is possible depending on the particular Syslog recipient(s) in use.

5.2.2 Memory Log Receiver

D-Link firewalls can act as log receivers with their built-in memories. When memory log receiver is enabled in the firewall, all events will be saved to the log file in the memory, and the most currently generated entries of the file can be displayed to the administrator upon requests. This log file storage is temporary, all contents of the file will be cleaned after reboot of the firewall, and there is no backup. Only one memory log receiver can be configured for a single firewall.

5.2.3 SMTP Event Receiver

A unique feature designed for IDS/IDP events logging and alerts is provided by D-Link firewalls, named as *SMTP Event Receiver*. Upon proper configuration, the firewall is able to log possible intrusions and notifies the administrator by sending e-mail(s) to specific e-mail address(es). For more information about this function, please refer to [19.5 SMTP Log Receiver for IDS Events](#).

CHAPTER 6

Maintenance

6.1 Firmware Upgrades

D-Link Firewalls can be upgraded with new firmwares to introduce new functionality and fix known problems. Make sure to regularly check on the D-Link support website for new firmware upgrades.



Example: Upgrading Firmware

This example describes how to upgrade a D-Link Firewall with a new firmware version.

WebUI :

1. Check Current Version

First of all, check which firmware version is currently running on the D-Link Firewall.

Status → **System**: Take note of the "Firmware Version" number seen under "System Status".

2. Download Firmware Upgrade

Go to the D-Link support website and navigate to the support section of your firewall model. Check if a upgrade to a newer firmare version than you are currently running on the firewall is available.

If a new version exists, download it and place it on your harddrive and take note of where you placed the new file.

3. Upgrade the Firewall Firmware

Go to the WebUI of your D-Link Firewall and navigate to **Tools** → **Upgrade** page in the toolbar. Under "Firmware Upload", click on the "Browse" button. Select the firmware upgrade file you recently downloaded from the D-Link support website.

Click on the "Upload Firmware" button and wait until the file is uploaded and further instructions are shown on the page.



Caution

DO NOT ABORT THE FIRMWARE UPLOAD PROCESS.

The firmware upload may take several minutes depending on the speed of your connection to the firewall.

6.2 Reset To Factory Defaults

There are three ways to reset the D-Link Firewall to its default firmware and configuration.

1. Reset To Factory Defaults From the WebUI

In the WebUI of the Firewall navigate to **Tools** → **Reset** page in the toolbar. Select **Reset to Factory Defaults**, confirm and wait for the revert process to complete.

2. Reset To Factory Defaults Via the Serial Console

Connect the serial cable and attach using terminal emulator software (if Windows is used, the communication accessory HyperTerminal can be used). Reset the firewall. Press any key when "Press any key to abort and load boot menu" message appear on the console. When the bootmenu appear select "Reset to factory defaults", confirm and wait for the revert process to complete.

The following procedure only applies to the DFL-800:

3. Reset To Factory Defaults Using the Reset Switch

Reset the firewall. Press and hold the "reset to factory defaults" button for 20 seconds. Wait for the revert process to complete and the firewall to start.

The following procedure only applies to the DFL-1600/2500:

3. Reset To Factory Defaults Using the Keypad and Display

Reset the firewall. Press any key on the keypad when the "Press keypad to Enter Setup" message appear on the display. Select "Reset firewall", confirm by selecting "yes" and wait for the revert process to complete.



Caution

DO NOT ABORT THE RESET TO FACTORY DEFAULTS PROCESS.

If aborted the firewall can cease to function properly.

After the reset process, the settings of the firewall will be permanently restored.

6.3 Backup Configuration

D-Link Firewalls configuration can be backed up to and restored at request. This could for instance be used to recall the "last known good" configuration when experimenting with different configuration setups.

To create a backup of the current running configuration:

WebUI :

Create and Download Backup Package

In the WebUI of the D-Link firewall navigate to the **Tools** → **Backup** page in the toolbar. Click "Download configuration", select a name for the backup snapshot and download the package.

To restore a backup configuration:

WebUI :

Restore Backup Package

In the WebUI of the D-Link firewall navigate to the **Tools** → **Backup** page in the toolbar. In the "Restore unit's configuration" subsection, use the browse functionality to locate the backup package. Click "Upload configuration" and when asked, choose to activate the configuration.



- The backup functionality ONLY include the firewall configuration. Dynamic information such as the DHCP server lease database or the Zone Defense blocking list will not be backed up. ■

CHAPTER 7

Advanced Settings

7.1 Overview

Advanced Settings contain various global settings for a firewall in terms of *packet size limits*, *connection timeouts*, *protocol parameters*, the *structural integrity tests* each packet shall be subjected to, etc.

Generally, the default values given in these sections are appropriate for most installations. But such option gives advanced installations a possibility to configure almost all aspects of the firewall.

WebUI :

In the WebUI, there is an **Advanced Settings** section located at:
System → **Advanced Settings**.

Most of the general configurable settings are available here. Other advanced settings for customizing specific functions of the firewall can be found in the configuration page inside the relevant sections.

One case that requires changes to the advanced settings is explained in 17.4, *Example: Enabling HTTP authentication via local user database*. Note that in this example, advanced settings in the firewall's **Remote Management** section need to be changed to resolve a TCP port number collision with HTTP authentication service.

Part V

Fundamentals

From both physical and logical perspectives, this part introduces the basic components of D-Link firewalls, which are the building blocks for security policies and advanced functions.

Topics in this part includes:

- [Logical Objects](#)
- [Interfaces](#)
- [Routing](#)
- [Date & Time](#)
- [DNS](#)
- [Log Settings](#)

CHAPTER 8

Logical Objects

Logical objects are basic network elements defined in the firewall, referring to the entities needed to be protected and also the untrusted resources and applications that should be monitored by the security policies.

8.1 Address Book

Like the contacts book which records people's name with one's phone number and email address, the address book in a Firewall is a list of symbolic names associated with various types of addresses, including IP addresses and ethernet MAC addresses. These items are fundamental elements heavily used in the firewall's configuration, such as specifying filtering fields for security policies. Therefore, choosing a descriptive and easily remembered name for each address item will greatly ease administration work. The administrator can use the names in each configuration task instead of filling in addresses every time, and in case of any modification to an address, only one point in the address book need to be changed.

8.1.1 IP address

To enable every entity receiving and sending data from or to a TCP/IP network, a network layer (OSI layer 3) *IP address* is required to associate with each point between the network entity and the physical link, that is an *interface*. In other words, each interface has a unique IP address in the

network to indicate its location.

The address book in D-Link firewalls allows administrators to name IP addresses either for a single *host*, a *network*, a *master/slave pair* used in high availability, or a *group* of computers or interfaces. An address "0.0.0.0/0" named as "***all-nets***" is used to denote all possible networks. Examples of "IP4Host/Network" are given below.

Authentication of users from an IP address object can be enabled on "*IP₄ Host/Network*" or "*IP₄ Address Group*" by attaching user names or user groups to the object. Once the firewall checks the traffic flow from an address object and finds the user name defined on it, it will prompt the user with authentication request according to **User Authentication Rules** (See [17 User Authentication](#)).



Example: Specifying an IP4 Host

The IP address "192.168.0.1" is defined for the local network interface named as "*lan_ip*".

WebUI :

Objects → Address Book → InterfaceAddresses → Add → IP4 Host/Network → General:

Enter the following and then click **OK**:

Name: lan_ip

IP Address: 192.168.0.1

(InterfaceAddresses is an Address Folder to group the interfaces' IP addresses)



Example: Specifying an IP4 Network

The local network "192.168.0.0/24" is defined as "lanet".

WebUI :

Objects → Address Book → InterfaceAddresses → Add → IP4 Host/Network → General:

Enter the following and then click **OK**:

Name: lanet

IP Address: 192.168.0.0/24



Example: Enabling User Authentication for an IP Object

A user group "users" is defined into the local network address "lannet" to create an authentication address object "lannet_users". For information of specifying the user group, please refer to [17.4 Scenario](#).

WebUI :

1. Specifying an IP4 Network object "lannet" as shown in last example.
2. **Objects** → **Address Book** → **Add** → **IP4 Address Group**
→ **General**:
Enter the following:
Name: lannet_users
Group members:
From the list **Available**, select the "lannet" object and put it into the **Selected** list.
Comments: Auth. "users" on lannet

→ **User Authentication**:
Enter the name of the user group and then click **OK**:
Comma-separated list of user names and groups: users

8.1.2 Ethernet address

An ethernet address, also know as a LAN address, a physical address, or a MAC (media access control) address, is a unique data-link layer (OSI layer 2) identifier of the network interface card, i.e. an ethernet adapter, which is used for sending the link-layer data frames. Users can also give a specific name to an ethernet address or an address group as explained in [8.1.1](#) above.

8.2 Services

Services are software programs using protocol definitions to provide various applications to the network users. Most applications rely on protocols located at OSI layer 7 – *Application layer* – to provide communication from

one user's program to other parties in a network. At this layer, other parties are identified and can be reached by specific *application protocol types* and corresponding parameters, such as *port numbers*. For example, the Web-browsing service HTTP is defined as to use the *TCP protocol* with destination *port 80*. Some of the other popular services at this layer include FTP, POP3, SMTP, Telnet, and so on. Beside these officially defined applications, user customized services can also be created in D-Link firewalls.

Services are simplistic, in that they cannot carry out any action in the firewall on their own. Thus, a service definition does not include any information whether the service should be allowed through the firewall or not. That decision is made entirely by the firewall's *IP rules*, in which the service is used as a filter parameter. For more information about how to use services in rules, please see [14 IP Rules](#).

8.2.1 Service Types

In D-Link firewalls, services can be configured via three options: *TCP/UDP*, *ICMP*, and *IP Protocol* service. A service is basically defined by a descriptive *name*, the *type* of the protocol, and protocol *parameters*. Different services can be united into one *Service Group* to simplify policy configuration, so that the administrators do not need to configure one rule for every service.

TCP and UDP based services

Service applications most commonly run on TCP or UDP, and are often associated with a well-known port number. In the firewall, they are defined by the type of protocol that the application uses, and the assigned port number or port range. For many services, a single *destination port* is sufficient. The HTTP service, for instance, uses TCP destination port 80, Telnet uses TCP 23, and SMTP uses TCP 25. In these cases, all ports (0-65535) will be accepted as source ports.

Multiple *ports* or *port ranges* may also be set, for instance, a service can be defined as having source ports 1024-65535 and destination ports 80-82, 90-92, 95. In this case, a TCP or UDP packet with the destination port being one of 80, 81, 82, 90, 91, 92 or 95, and the source port being in the range 1024-65535, will match this service.

**Example:** Specifying a TCP service -- HTTP

In this example, the service HTTP for connecting web servers is defined. As explained previously, HTTP uses TCP destination port 80.

WebUI :

Objects → Services → Add → TCP/UDP:

Enter the following and then click **OK**:

General

Name: HTTP

Type: TCP

Source: 0-65535

Destination: 80

ICMP based services

Internet Control Message Protocol (ICMP), is a protocol integrated with IP for error reporting and transmitting control information. The *PING* service, for example, uses ICMP to test an Internet connectivity. ICMP message is delivered in IP packets, and each message is a separate protocol having its own format. Its content changes depending on the *Message Type & Code*.

The ICMP message types that can be configured in D-Link firewalls along with the various codes are listed as follows:

- **Echo Request** – sent by PING to a destination in order to check connectivity.
- **Destination Unreachable** – the source is told that a problem has occurred when delivering a packet. There are codes from 0 to 5 for this type:
 - Code 0. Net Unreachable
 - Code 1. Host Unreachable
 - Code 2. Protocol Unreachable
 - Code 3. Port Unreachable
 - Code 4. Cannot Fragment
 - Code 5. Source Route Failed

- **Redirect** – the source is told that there is a better route for a particular packet. Codes assigned are as follows:
 - Code 0. Redirect datagrams for the network
 - Code 1. Redirect datagrams for the host
 - Code 2. Redirect datagrams for the Type of Service and the network
 - Code 3. Redirect datagrams for the Type of Service and the host
- **Parameter Problem** – identifies an incorrect parameter on the datagram.
- **Echo Reply** – the reply from the destination which is sent as a result of the Echo Request.
- **Source Quenching** – the source is sending data too fast for the receiver, the buffer has filled up.
- **Time Exceeded** – the packet has been discarded as it has taken too long to be delivered.



Example: Adding a custom ICMP service

A custom ICMP service is added and can be used in security policies.

WebUI :

Objects → **Services** → **Add** → **ICMP Service**

→ **General:**

Enter a Name for the new ICMP service.

→ **ICMP Parameters**

Select the ICMP **type** and specify the **codes** for the service.

(If the **All ICMP Message Types** option is selected, this service will match all 256 possible ICMP Message Types.)

Click **OK**.

User-defined IP protocol service

Services that run over IP and perform application/transport layer functions can be defined by *IP protocol numbers*. IP can carry data for a number of different protocols. These protocols are each identified by a unique IP protocol number specified in a field of the IP header, for example, ICMP, IGMP, and EGP have protocol numbers 1, 2, and 8 respectively. The currently assigned IP protocol numbers and references are published on the web site of Internet Assigned Numbers Authority (IANA):

<http://www.iana.org/assignments/protocol-numbers>

Similar to TCP/UDP port range described previously, a range of IP protocol numbers can be used to specify multiple applications for one service.



Example: Adding a service that matches the GRE protocol

(For more information about GRE, please refer to [22.2 PPTP/L2TP](#))

WebUI :

Objects → **Services** → **Add** → **IP Protocol Service**

General

Enter the following and then click **OK**:

Name: GRE

IP Protocol: 47

Service Group

The services defined in the above options can be grouped in order to simplify security policy configuration. Consider a web server using standard HTTP as well as SSL encrypted HTTP (HTTPS, refer to [22.3 SSL/TLS\(HTTPS\)](#)). Instead of having to create two separate rules allowing both types of services through the firewall, a service group named, for instance, "Web", can be created, with the HTTP and the HTTPS services as group members (shown in the example below).



Example: Specifying a "Web" service group

WebUI :

Follow the steps outlined below:

1. Adding a TCP service object "HTTP" with port 80.
2. Adding a TCP service object "HTTPS" with port 443.

3. Objects → Services → Add → Service Group

General

Name: Web

Pick "HTTP" and "HTTPS" from **Available** list and put them into **Selected** list.

Click **OK**.

8.2.2 Error Report & Connection Protection

ICMP error message

ICMP error messages provide feedback about problems in the communication environment, e.g. when an IP packet cannot reach its destination. However, ICMP error messages and firewalls are usually not a very good combination; the ICMP error messages are initiated at the destination host (or a device within the path to the destination) and sent to the originating host. The result is that the ICMP error message will be interpreted by the firewall as a new connection and dropped, if not explicitly allowed by the firewall rule-set. Allowing any inbound ICMP message to be able to have those error messages forwarded is generally not

a good idea, since it may cause the protected network vulnerable to many types of attacks, e.g. DoS (Denial of Service) in particular.

To solve this problem, D-Link firewalls can be configured to pass an ICMP error message only if it is related to an existing connection of a service.

SYN flood protection (SYN Relay)

A mechanism called "SYN Relay" can be enabled in the firewall to protect the destination addresses used by a service from SYN flooding.

The SYN flood attack is launched by sending TCP connection requests faster than a machine can process them. The attacker sends SYN request to a server with spoofed source address, which will never reply to the server's SYN/ACK. Each SYN request fills in a new TCP connection into the server's connection table; when all the connections in the table are waiting for replies and the table is full, the server will not accept any new coming request. The requests from legitimate users are then ignored.

The "SYN Relay" mechanism counters the attacks by hiding the protected server behind the firewall. The firewall receives SYN request and makes sure that the connection is valid (that is, the SYN/ACK can be replied from the source) before sending a SYN packet to the server. If after a certain time, no ACK is received by the firewall, the connection is aborted.

Application Layer Gateway (ALG)

An application layer gateway can be specified to handle different services. More information can be found in [18 Application Layer Gateway \(ALG\)](#). For an ALG enabled service, the maximum numbers of sessions that are permitted by using this service can be defined.

8.3 Schedules

Scheduling is a way to create timing constraints on the firewall's rules. It enables the user to define a certain time period, in the format of year–date–time, which will only activate the rules at the designated times. Any activities outside of the scheduled time slot will not follow the rules and will therefore likely not be permitted to pass through the firewall. The schedules can be configured to have a start time and stop time, as well as creating different time periods in a day.



Example: An office-hour schedule

An organization may only want the internal network users to access the Internet during work hours, and expect this constraint to be valid for one year. Therefore, one may create a schedule to restrict the firewall to allow traffic Monday-Friday, 8AM-5PM only, starting from a certain date and time, say 2005-04-01 00:00:00, to an end date. During the non-work hours, the firewall will not allow the access.

WebUI :

Objects → Schedule Profiles → Add → Schedule Profile:

General

Name: Enter a name for this schedule, e.g. "office-hour".

Define a time period by checking the boxes.

Start Date: Fill in the start time in a format of "yyyy-mm-dd hh:mm:ss" or click the calendar icon and choose a date from the pop-up window.

End Date: (same as "Start Date" above)

and then click **OK**.

8.4 X.509 Certificates

D-Link firewalls support certificates that comply with the ITU-T X.509 international standard. This technology use an X.509 certificate hierarchy with public-key cryptography (See [20.2, *Introduction to Cryptography*](#)) to accomplish key distribution and entities authentication.

8.4.1 Introduction to Certificates

A certificate is a digital proof of identity. It links an identity to a public key for establishing whether a public key truly belongs to the supposed owner. Thus, it prevents data transfer interception by any ill-intending third-party, who may post a phony key with the name and user ID of an intended recipient. A certificate consists of the following:

- *A public key*: The "identity" of the user, such as name, user ID, etc.
- *Digital signatures*: A statement that tells the information enclosed in the certificate has been vouched for by a Certificate Authority (CA).

Binding the above information together, a certificate is a public key with identification forms attached, coupled with a stamp of approval by a trusted party.

Certification Authority

A certification authority (CA) is a trusted entity that issues certificates to other entities. The CA digitally signs all certificates it issues. A valid CA signature in a certificate verifies the identity of the certificate holder, and guarantees that the certificate has not been tampered with by any third party.

A certification authority is responsible for making sure that the information in every certificate it issues is correct. It also has to make sure that the identity of the certificate matches the identity of the certificate holder.

A CA can also issue certificates to other CAs. This leads to a tree-like certificate hierarchy. The highest CA is called the root CA. In this hierarchy, each CA is signed by the CA directly above it, except for the root CA, which is typically signed by itself.

A certification path refers to the path of certificates from one certificate to another. When verifying the validity of a user certificate, the entire path

from the user certificate up to the trusted root certificate has to be examined before establishing the validity of the user certificate.

The CA certificate is just like any other certificates, except that it allows the corresponding private key to sign other certificates. Should the private key of the CA be compromised, the whole CA, including every certificate it has signed, is also compromised.

Validity Time

A certificate is not valid forever. Each certificate contains the dates between which the certificate is valid. When this validity period expires, the certificate can no longer be used, and a new certificate has to be issued.

Certificate Revocation Lists (CRL)

A certificate revocation list (CRL) contains a list of all certificates that has been cancelled before their expiration date. This can happen for several reasons. One reason could be that the keys of the certificate have been compromised in some way, or perhaps that the owner of the certificate has lost the rights to authenticate using that certificate. This could happen, for instance, if an employee has left the company from whom the certificate was issued.

A CRL is regularly published on a server that all certificate users can access, using either the LDAP or HTTP protocols.

Certificates often contain a CRL Distribution Point (CDP) field, which specifies the location from where the CRL can be downloaded. In some cases certificates do not contain this field. In those cases the location of the CRL has to be configured manually. See [22.1.4](#), *LDAP*.

The CA updates its CRL at a given interval. The length of this interval depends on how the CA is configured. Typically, this is somewhere between an hour to several days.

Trusting Certificates

When using certificates, the firewall trusts anyone whose certificate is signed by a given CA. Before a certificate is accepted, the following steps are taken to verify the validity of the certificate:

- Construct a certification path up to the trusted root CA.

- Verify the signatures of all certificates in the certification path.
- Fetch the CRL for each certificate to verify that none of the certificates have been revoked.

Identification Lists

In addition to verifying the signatures of certificates, D-Link firewalls also employ identification lists (See 22.1.4, *Identification Lists (IDLists)*). An identification list is a list naming all the remote identities that are allowed access through a specific VPN tunnel, provided the certificate validation procedure described above succeeded.

8.4.2 X.509 Certificates in D-Link Firewall

X.509 certificates can be uploaded to the D-Link Firewall for use in IKE/IPSec authentication, webauth etc. There are two types of certificates that can be uploaded, self signed certificates and remote certificates belonging to a remote peer or CA server.



Example: Uploading a Certificate to a D-Link Firewall

This example describes how to upload a X.509 certificate to a D-Link Firewall. The certificate may either be self-signed or belonging to a remote peer or CA server.

WebUI :

Upload Certificate

Objects → **X.509 Certificates** → **Add** → **X.509 Certificate**:

Enter the following:

Name: Name of the certificate.

Options

Select one of the following:

- Upload self-signed X.509 Certificate
- Upload a remote certificate

Then click **OK** and follow the instructions on the screen.

CHAPTER 9

Interfaces

Physical interfaces are the doorways of the network connections. It allows the network traffic to enter into or go out of the network areas with which it connects. In order to control the traffic on both in and out directions and protect the local network, security rules in the firewall is bound to all relevant interfaces.

9.1 Ethernet

Ethernet is one of the Local Area Network (LAN) architectures served as the basis for the IEEE 802.3 standard, which specifies the physical and lower software layers. It is one of the most widely implemented LAN standards. This section presents the concepts of Ethernet interface. Some of the commonly used protocols that run on Ethernet are introduced in sections [9.2 VLAN](#) and [9.4 PPPoE](#) in this chapter, others like IPsec, PPTP, L2TP, and ARP are covered later in the document.

9.1.1 Ethernet Interfaces

An Ethernet interface represents a physical Ethernet adapter used in the firewall. The configuration of an Ethernet interface involves the assignment of an IP address and other parameters, to make the interface accessible to the network layer.

When installing a D-Link firewall, all supported Ethernet adapters in the

firewall will be enumerated and configured during the local console setup process. Each physical Ethernet adapter will become an Ethernet interface and a name will be given in the firewall configuration. Administrators can customize the descriptive name and change the IP addresses of an interface after the primary installation.

9.1.2 Ethernet Interfaces in D-Link Firewalls

Configuration of an Ethernet interface mainly includes specifying the name and the addresses. An IP address is bound to every interface that may be used to ping the firewall, remotely control it, and be set by the firewall as source address for dynamically translated connections. An additional IP address can be published on an interface using ARP to simulate the effect of an interface having more than one IP (See [9.6 ARP](#)). Moreover, administrators can apply dynamic address assignment to a network by enabling DHCP client on the corresponding interface (See [9.3 DHCP](#)).

As advanced features, *High Availability(HA)* & *Transparency* can be implemented on the basis of firewall interfaces.

The HA enabled interfaces share one common IP address and each has a private IP address to uniquely identify one cluster node. The private IP is derived from the **HA IP4 Address Pair** configured in the **Address Book** object (See [XIII High Availability](#) for more information about HA cluster scenarios).

When setting up an interface to use transparent mode, the firewall will act as a layer 2 switch and screen the traffic going through that interface without modifying the source or destination address information. Both sides of the communication will be unaware of the presence of the firewall. For transparent mode configuration on interfaces, please refer to [27 Transparency](#).



- In the firewall, there are two logical interfaces named as "core" and "any" respectively. "core" locates at the heart of the firewall, all traffic from the physical interfaces are forwarded to "core" to be controlled by security policies. "any" represents all possible interfaces including "core". ■

**Example:** A LAN interface configuration

The interface connected to LAN (or one of the LANs) is configured with "lan_ip", "lannet", and the default gateway address "lan_gate".

WebUI :

1. Specifying the **IP4 Host** – "lan_ip" and "lan_gate", and an **IP4 Network** – "lannet" in the **Objects**.
(See examples in [8.1 Address Book](#))
2. **Interfaces** → **Ethernet**:
Click the item for LAN interface
→ **General**:
Name: Define or change the name of the interface in the edit box
IP Address: Select "lan_ip" from the dropdown list.
Network: Select "lannet" from the dropdown list.
Default Gateway: Select "lan_gate" from the dropdown list.
and then click **OK**.
3. Optional settings:
→ **General**:
Enable DHCP Client check box
Enable Transparent Mode check box

→ **Hardware Settings**:
(The network adapter's hardware settings can be adjusted here.)
Media – Specifies if the link speed should be auto-negotiated or locked to a static speed.
Duplex – Specifies if duplex should be auto-negotiated or locked to full or half duplex.

→ **Advanced**:
Automatic Route Creation check boxes
Route Metric edit box
(By checking these options and specifying the metric value, the interface configured here will be added into the **Main Routing Table** as routes for destination address information. The default metric value is 100.)
High Availability: Private IP Address selection.

9.2 Virtual LAN (VLAN)

Virtual Networking is the ability of network appliances to manage the logical network topologies on top of the actual physical connections, allowing arbitrary segments within a network to be combined into a logical group. Since the flexibility and the ease of network control provided by the logical topologies, virtual networking has become one of the major areas in the internetworking.

D-Link firewalls are fully compliant with IEEE 802.1Q specification for Virtual LANs, featured by defining virtual interfaces upon the physical Ethernet interface. Each virtual interface is interpreted as a logical interface by the firewall, with the same security policies control and configuration capabilities as regular interfaces.

9.2.1 VLAN Infrastructure

A Local Area Network (LAN) is a broadcast domain, that is, a section of the network within whose boundaries any broadcast traffic is delivered to all end-nodes. When the LAN environment grows bigger, the support of broadcast or multicast applications that flood packets throughout the network costs considerable waste of bandwidth, since packets are often forwarded to nodes that do not require them.

Virtual LAN (VLAN) allows a single physical LAN to be partitioned into several smaller logical LANs which are different broadcast domains. It limits the size of the broadcast domain for each logical LAN, saves the broadcast cost of the bandwidth to optimize the performance and resource allocation, and also divides larger LANs into several independent security zones to add security control points. Devices located in the same LAN can communicate without the awareness of the devices in other virtual LANs. This is ideal for separating industrial departments from physical topology to different function segments.

A simple infrastructure of VLAN is shown in Figure 9.1. In this case, a D-Link firewall is configured to have 2 VLAN interfaces. Now, although the clients and servers are still sharing the same physical media, Client A can only communicate with Server D and the firewall since they are configured

to the same VLAN, and Client B can only communicate with Server C through the firewall.

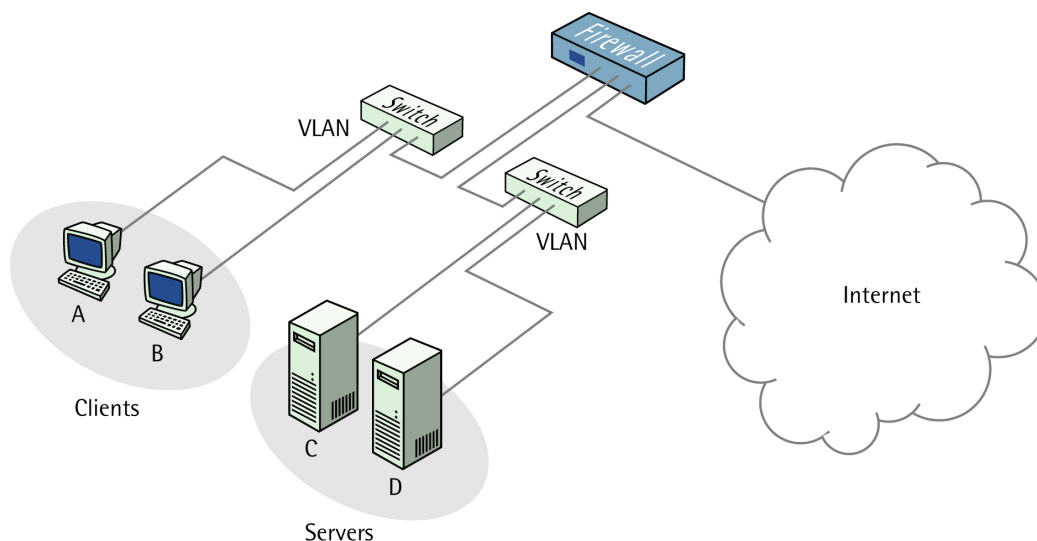


Figure 9.1: A VLAN Infrastructure.

9.2.2 802.1Q VLAN Standard

The IEEE 802.1Q standard defines the operation of VLAN devices that permit the definition, operation and administration of Virtual LAN topologies within a LAN infrastructure.

802.1Q specification establishes a standard method for tagging Ethernet frames with VLAN membership information. As defined in the standard, a 4-byte tag is appended to the Ethernet frame containing one part for VLAN frame type indicator (0x8100), one VLAN identifier (VID), and some control information (shown in Table 9.1).

There are 12 bits for VID within each 4-byte tag. With these 12 bits of identifier, there could be up to 4096 VLANs on a physical network. However, all ones are reserved and all zeros indicate no VLAN association. All other identifiers can be used to indicate a particular VLAN.

bytes									
8	6	6	4				2	46 to 1500	4
Pre- amble	Dest.	Sou- rce	32 bits				Len- gth	Data	CRC
			16	3	1	12			
			VLAN Type Indicator (0x8100)	Pri- ority	CFI	VID			
VLAN Tag									

Table 9.1: 802.1Q Standard Ethernet Frame.

9.2.3 VLAN Implementation

Comply to 802.1Q standard, the D-Link firewall implement VLAN by defining one or more VLAN interfaces on it using a unique VID for each VLAN. When Ethernet frames are received by the firewall, they are examined for the VID. If a VID is found, and a matching VID interface has been defined, the firewall will be able to recognize the membership and destination of that VLAN communication.

VLANs in D-Link firewalls are useful in several different scenarios, for instance, when firewall filtering is needed between different departments in an organization, or when the number of interfaces needs to be expanded.

**Example:** Configure a VLAN Interface in D-Link Firewall

This example shows how to configure a VLAN interface.

WebUI :

1. Create VLAN interface.

Interfaces → **VLAN** → **Add** → **VLAN**:

Enter the following:

General

Name: Type a name for the VLAN interface.

Interface: Select the Ethernet interface to use.

VLAN ID: Select a suitable VLAN ID. Two VLANs cannot have the same VLAN ID if they are defined on the same Ethernet interface. (The same ID will have to be used on the terminating side.)

Address Settings

IP Address: Select the IP address this VLAN interface should use. If not configured, the IP of the Ethernet interface will be used. (Optional)

Network: Select the network for this VLAN interface. (Optional)

Default Gateway: Select the default gateway for this VLAN interface. (Optional)

Then click **OK**

9.2.4 Using Virtual LANs to Expand Firewall Interfaces

Virtual LANs are excellent tools for expanding the number of interfaces in D-Link Firewalls. The D-Link Firewalls with gigabit Ethernet interfaces can easily be expanded with 16 new interfaces by using a 16-port Ethernet switch with gigabit uplink port and Virtual LAN support.

The process outlined below describes the steps required to perform an interface expansion. Please note that the specific configuration of switch and firewall is highly model dependent and outside the scope of this documentation.

- Connect the gigabit uplink port of the switch to one of the gigabit

interfaces on the firewall.

- Create 16 Virtual LANs in the firewall, named, for instance, vlan01 to vlan16, each with a unique VLAN ID
- In the switch, map each VLAN ID to a switch port, and make sure the uplink port is configured as a trunk port for all the VLAN IDs.
- Each port of the switch will now be seen as a logical interface in the firewall. Thus, traffic entering the switch through, for instance, port 12 will be received by interface vlan12 in the firewall.

In the example above, a gigabit uplink port on the switch and a gigabit interface on the firewall was used. Gigabit interfaces are not a requirement from a functionality perspective; any type of interface would have worked. However, from a performance perspective, gigabit interfaces are recommended. Remember that one single Ethernet link is used to carry all traffic from the 16 switch ports, each with an interface link speed of 100 Mbps.

9.3 DHCP

Short for *Dynamic Host Configuration Protocol*, DHCP is the third-generation host configuration protocol for TCP/IP, which is based directly on the BOOTP (Boot Protocol). It is used for automatic allocation of network addresses and configurations to newly attached hosts.

The purpose of using DHCP is to reduce the work necessary to administer a large IP network. There is software mechanism to keep track of IP addresses rather than requiring an administrator to manage the task. This means that a new computer can be added to a network without the hassle of manually assigning it a unique IP address. D-Link Firewall appliance can act as either a DHCP client, a server, or a relay through the interfaces. DHCP server and relay functions are covered in [26](#), DHCP Server & Relay.

9.3.1 DHCP Client

The DHCP client broadcasts message to locate a DHCP server(or servers) and receives an IP address dynamically from a DHCP server for its physical interface. A DHCP client may receive offers from multiple DHCP servers

and usually accepts the first offer it receives. Clients can renew or release their IP address assignment during the lease period.



Example: Configuring the firewall as a DHCP client

To enable the firewall acting as a DHCP client and locate external DHCP server(s) and receive address information dynamically, follow the steps below:

WebUI :

Interfaces → **Ethernet:**

Click the interface that is connected to the external network and to be used as DHCP client.

→ **General:**

Adjust the name and addresses of the interface.
(See the example in [9.1 Ethernet](#))

Check the **Enable DHCP Client** check box.
and then Click **OK**.

9.4 PPPoE

Point-to-Point Protocol over Ethernet (PPPoE) relies on the two widely accepted standards: Point-to-Point protocol (PPP) and Ethernet. It is used for connecting multiple users on an Ethernet network to the Internet through a common serial interface, such as a single DSL line, wireless device or cable modem. All the users over the Ethernet share a common connection, meanwhile, the access control and service can be done on a per-user basis.

Today, many large Internet server providers (ISPs) require customers to connect through PPPoE to their broadband service. Using PPPoE, the provider can easily perform the following functions for each user:

- Support security and access-control – username/password authentication is required. The provider can track IP address to a specific user.
- Automatic IP address allocation for PC users (similar to DHCP [9.3](#)).

IP addresses provisioning can be per user groups.

9.4.1 PPP

Point-to-Point Protocol (PPP), is a protocol for communication between two computers using a serial interface, for instance, a dialup connection where a personal computer is connected by telephone line to a server.

The ISP provides the user with a PPP connection so that the provider's server can respond to the user's requests, pass them on to the Internet, and forward requested Internet responses back to the user. Relative to the OSI reference model, PPP provides *Layer 2 (data-link layer)* service.

At Layer 2, PPP defines an encapsulation mechanism to support multi-protocol packets to travel through IP networks. It starts with a *Link Control Protocol (LCP)* for link establishment, configuration and testing. Once the LCP is initialized, one or several *Network Control Protocols (NCPs)* can be used to transport traffic for a particular protocol suite, so that multiple protocols can interoperate on the same link, for example, both IP and IPX traffic can share a PPP link.

Authentication is available as an option for PPP communications. The authentication protocols that PPP currently supports include:

- Password Authentication Protocol (PAP)
- Challenge Handshake Authentication Protocol (CHAP)
- Microsoft CHAP version 1
- Microsoft CHAP version 2

If authentication is used, at least one of the peers has to authenticate itself before the network layer protocol parameters can be negotiated using NCP. During the LCP and NCP negotiation, optional parameters such as encryption, can be negotiated. When LCP and NCP negotiation is done, IP datagrams can be sent over the link. More about the *application* and *security* of PPP can be found in section [22.2 PPTP/L2TP](#).

9.4.2 PPPoE Client Configuration

D-Link firewalls allow users a secure and easy-to-manage connection to the ISP.

PPPoE interface

Since the PPPoE protocol runs PPP over Ethernet, the firewall needs to use one of the normal Ethernet interfaces to run the PPPoE tunnel over. Each PPPoE Tunnel is interpreted as a logical interface by the firewall, with the same filtering, traffic shaping and configuration capabilities as regular interfaces.

The network traffic coming from the PPPoE tunnel will be transferred to the firewall ruleset for evaluation. The source interface of the network traffic is referred to the name of the associated PPPoE Tunnel in the firewall. The same is true for traffic coming from the opposite direction, that is, going into a PPPoE tunnel. Furthermore a Route has to be defined, so the firewall knows what IP addresses should be accepted and sent through the PPPoE tunnel. PPPoE can use a service name to distinguish between different servers on the same Ethernet network.

IP address information

PPPoE uses automatic IP address allocation which is similar to DHCP. When the firewall receives this IP address information from the ISP, it needs to store it in a network object with symbolic host/network names, in order to establish the PPP connection.

User authentication

If user authentication is required by the ISP, we can set in the firewall the user name and password for logging on to the PPPoE server.

Dial on demand

If dial-on-demand is enabled, the PPPoE connection will only be up when there is traffic on the PPPoE interface. It is possible to configure how the firewall should sense activity on the interface, either on outgoing traffic, incoming traffic or both. Also configurable is the time to wait with no activity before the tunnel is disconnected.



Example: A PPPoE Client configuration

This example describes how to configure a PPPoE client. The PPPoE client is configured on the WAN interface and all traffic should be routed over the PPPoE tunnel.

WebUI :

PPPoE Client

We will configure the PPPoE client on the WAN interface.

Interfaces → **PPPoE Tunnels** → **Add** → **PPPoE Tunnel:**

Enter the following:

Name: PPPoEClient

Physical Interface: WAN

Remote Network: 0.0.0.0/0 (all-nets, as we will route all traffic into the tunnel)

Service Name: If your service provider has provided you with a service name, enter the service name here.

Username: The username provided to you by your service provider.

Password: The password provided to you by your service provider.

Confirm Password: Retype the password.

Authentication

It is possible to specify exactly which protocols the PPPoE client should try to authenticate with. We keep the default settings for authentication.

Dial-on-demand

Enable Dial-on-demand: Disable

Advanced

If "Add route for remote network" is enabled, a new route will be added for this interface.

Then click **OK**

9.5 Interface Groups

Similar to logical object group, multiple interfaces can be grouped together in the firewall to apply to a common policy. An interface group can consist of regular Ethernet interfaces, VLAN interfaces, or VPN Tunnels (see [22](#)). All members of an interface group do not need to be interfaces of the same type. This means that an interface group can be built from, for instance, two Ethernet interfaces and four VLAN interfaces.



Example: An Interface Group Example

This example describes how to configure an interface group object.

WebUI :

- **Create Interface Group**

Interfaces → **Interface Groups** → **Add** → **Interface Group**:

Enter the following:

Name: testifgroup

Security/Transport Equivalent: If enabled, the interface group can be used as a destination interface in rules where connections might need to be moved between the interfaces. Examples of such usage can be Route Fail-Over and OSPF (see [10.3.3](#)) scenarios.

Interfaces: Select the interfaces that should be a part of the group. Then click **OK**

- **Use the Interface Group**

An interface group can be used in various object configurations. For example, IP rules and user authentication rules can use interface groups.

9.6 ARP

Address Resolution Protocol (ARP) is a network protocol, which maps a network layer protocol address to a data link layer hardware address. For example, ARP is used to resolve IP address to the corresponding Ethernet address. It works at the OSI Data Link Layer (Layer 2) and is encapsulated by Ethernet headers for transmission.

A host in an Ethernet network can communicate with another host, only if it knows the Ethernet address (MAC address) of that host. The higher level protocols like IP uses IP addresses. These are different from the lower level hardware addressing scheme like MAC address. ARP is used to get the Ethernet address of a host from its IP address.

When a host needs to resolve an IP address to Ethernet address, it broadcasts an ARP request packet. The ARP request packet contains the source MAC address and the source IP address and the destination IP address. Each host in the local network receives this packet. The host with the specified destination IP address, sends an ARP reply packet to the originating host with its MAC address.

9.6.1 ARP Table

The ARP Table is used to define static ARP entries (static binding of IP addresses to hardware addresses) or to publish IP addresses with a specific hardware address.

Static ARP items may help in situations where a device is reporting incorrect hardware address in response to ARP requests. Some workstation bridges, such as radio modems, have such problems. It may also be used to lock an IP address to a specific hardware address for increasing security or to avoid denial-of-service effects if there are rogue users in a network. Note however that such a protection only applies to packets being sent to that IP address, it does not apply to packets being sent from that IP address.

Publishing an IP address using ARP can serve two purposes:

- To aid nearby network equipment responding to ARP in an incorrect manner. This area of use is less common.
- To give the impression that an interface of the firewall has more than one IP address.

To accomplish the above, the firewall provides responses to ARP requests concerning the IP addresses in published ARP items. The latter purpose is useful if there are several separate IP spans on a single LAN. The computers on each IP span may then use a gateway in their own span when these gateway addresses are published on the firewall interface.

Another area of use is publishing multiple addresses on an external interface, enabling the firewall to statically address translate communication to these addresses and send it onwards to internal servers with private IP addresses.

The difference between XPublish and Publish is that XPublish "lies" about the sender hardware address in the Ethernet header; this is set to be the same as the published hardware address rather than the actual hardware address of the network adapter.

If a published hardware address is the same as the hardware address of the network adapter, it will make no difference if you select Publish or XPublish; the net result will be the same.

■  **Note** ■

■ In the ARP section, addresses may only be published one at a time. The Routes section on the other hand, can handle publishing entire networks using [10.8 Proxy ARP](#). ■

■  **Note** ■

■ For published IP addresses to work correctly it might be necessary to add a new route. (See [10 Routing](#)) If an additional address is added for an interface, the core interface should probably be specified as the interface when configuring the route. ■



Example: An ARP Example

This example describes how to add an extra IP address to a Ethernet or VLAN interface using ARP publish.

WebUI :

- **Create a ARP Table entry**

Interfaces → **ARP Table** → **Add** → **ARP Entry:**

Enter the following:

Mode: Publish

Interface: Select the interface that should have the extra IP address

IP Address: Specify the IP address to add to the above interface.

MAC: Leave it at 00-00-00-00-00-00 to use the MAC address of the interface.

Then click **OK**

CHAPTER 10

Routing

10.1 Overview

Routing is a major role in the network layer (OSI layer 3), which determines how to transport packets from the initiating host to the desired receiving end.

The devices functioning at the network layer, such as routers or firewalls, perform routing to achieve two tasks primarily, the *Path Determination* and the *Packet Switching*.

Path determination

Before any packet can be sent from the sender to the receiver, a *path* need to be determined for the packet to travel through. Located in the heart of any routing capable device, like a firewall or a router, is the *routing table*, a map that provides all the path selections. Each entry in this mapping table describes an available route.

The definition of the *route* here is the connection that links the two communication ends and also all the intermediate routing devices. The description of route inside the routing table indicates the address of the receiver, and where is the next stop(*hop*) the packet should go to get one step closer to its destination, since in the network circumstance, it is common to have more than one device sitting along the way. These contents are stored in the table as different

fields, such as *Interface*, *Network*, *Gateway*, *Destination*, etc.

When a packet arrives at a router, it refers to the routing table to make path determination. The router compares the destination address of the packet with the entries it has in the routing table, and finds out the associated interface and next hop from the matching route to forward the packet. The paths stored in the table are computed by certain *routing algorithm* defined for the router, which always tries to make the "best" choice. The "best" means a path selection having the "least cost" for transporting. In practice, the concern of "cost" are normally the *bandwidth*, *path length(hops)*, *average delay*, and etc., which are introduced in [10.3.2 Routing metrics](#).

Routing algorithm is also responsible for keeping the routing table up to date, so that the router can obtain correct path information for every decision. The two most prevalent classes of routing algorithms are covered in the next section.

Packet switching

After a path is chosen, the packet switching function takes control of how the packet is actually moved. According to the information of the selected route, the firewall/router rewrites the physical address of the packet to the address of the next hop, and forwards the packet to the next hop with the destination IP address unchanged. In a real-life scenario, many firewalls/routers may come into play during the packet forwarding process, each of them delivers the packet to its nearby neighbor until the packet finally arrives at the receiving host.

10.2 Routing Hierarchy

In a complex network environment, as the number of routers becomes large, the domain of routing is often divided into different areas to provide better scalability. Routers reside under the same administrative control are aggregated into one region called "*autonomous system (AS)*".

An *AS* can be, for example, all computer networks owned by a university or a company's private network. The organization is able to run and administer its network with its own policies and preferable routing algorithm independently, while still being able to connect to the "outside"

world.

Routers inside an AS run the same routing algorithm and they only need to know the topology of the area. There are special *gateway routers* in the ASs that are responsible for routing packets from internal area to various outside ASs. Gateway routers run *inter-AS routing* algorithm to determine the paths toward the destinations locating in other ASs. Intra-AS routers all maintain relationships with the gateway router to route packets out. The most prevalent intra-AS (interior gateway) routing algorithms are covered in the next section.

10.3 Routing Algorithms

A routing algorithm is an operator of the routing table. In the internetworking environment, there are typically several paths between two communication entities. The task of the routing algorithm is that given a set of intermediate routing devices with different links connecting them, the algorithm calculates a "best" path for two ends to communicate and appends the path into the table.

In case of a device failure (a down link) in a selected path or other problems that make the path unreachable, the algorithm selects the next best route and updates the routing table to maintain the content of the table correct.

Many routing algorithms have been proposed and each features different design for different goals. The most widely implemented algorithms can be categorized into two classes: *Static Routing* & *Dynamic Routing*.

10.3.1 Static Routing

Static routing is a term used to refer to the manually configuration of the routing table. The network administrators need to plan the routing table, and manually add every necessary route and related information into the table for successful packet forwarding. Any change on one path would require the administrator to update the information in every affected router.

As a result, the administration work in a large scale network environment can be very cumbersome and error prone. In the case that a route is not properly configured into the routing table, the router looks up in the table to make path determination and no suitable route can be found, it will

simply drop the packet. Therefore, static routing is often used to make the minimal set of routes to reach directly connected networks only.

10.3.2 Dynamic Routing

Complementing to static routing algorithm, *Dynamic Routing* adapts to changes of network topology or traffic loads automatically. It learns all the directly connected networks first, and gets further routes information from other routers that are running the same algorithm. The algorithm then sorts the routes it collected, selects the most suitable route for a destination it has learned, appends the route into its routing table, and distributes this information to other routers.

Dynamic routing responses to routing updates on the fly and is more susceptible to problems such as routing loops. In the Internet, two types of dynamic routing algorithm are typically used: a *Distance Vector(DV)* algorithm & a *Link State(LS)* algorithm. How it decides the "best" route and shares the update information with other routers depends on the type of the algorithm applied.

Distance vector algorithm

Distance vector (DV) algorithm is a decentralized routing algorithm, computing the "best" path in a distributed manner. Each router computes the costs of its own attached links, and shares the route information only with its neighbor routers. The router gradually learns the least-cost path by iterative computation and knowledge exchange with its neighbors.

The *Routing Information Protocol(RIP)* is a well-known DV algorithm. RIP sends update message regularly, and reflects the routing changes in the routing table. Its path determination is based on the length of the path – the number of intermediate routers, or the so-called *hops*. After updating its own routing table, the router immediately begins transmitting its entire routing table to its neighbor routers to inform the change.

Link state algorithm

Different from the DV algorithms, Link state (LS) algorithm enables the routers to keep routing tables that reflect the topology of the entire network, a global view of the routing information. As defined in this algorithm, each router *broadcasts* its attached links and link costs to all the other routers in the network. A router, upon receiving broadcasts from the

rest, runs the LS algorithm and can calculate a same set of least-cost paths as all the other routers. Any change of the link state will be sent everywhere in the network, so that all routers keep the same routing table information.

Open Shortest Path First(OSPF) is a commonly used LS algorithm. An OSPF enabled router identifies the routers and subnets that are directly connected to it first. Then, it broadcasts the information to all the other routers. Each router uses the information it received to build a table of what the whole network looks like. With an complete routing table at hand, each router can identify the subnetworks and routers that lead to any specific destination. The OSPF routers only broadcast updated information when there is any change instead of the whole table.

OSPF depends on various metrics for path determination, including hops, bandwidth, load, delay, and so on. User customized criteria is also allowed to be defined for the algorithm, which provides the network administrators greater control over the routing process. More details about OSPF algorithm are covered in [10.3.3 OSPF](#).

Comparison

Link state algorithm, because of its global link state information maintained everywhere in a network, has high degree of *configuration control* and *scalability*. It responses to changes by broadcasting only the updated information to all the others, and hence providing *faster convergence* and *smaller possibility of routing loops*. OSPF can also operate within a hierarchy, while RIP has no knowledge of subnetwork addressing. On the other hand, OSPF demands relatively higher cost, i.e. more CPU power and memory, than RIP, therefore, can be more expensive to implement.

D-Link firewalls deploy OSPF as the dynamic routing algorithm.

Routing metrics

Routing metrics(the costs) are the criterion a routing algorithm uses to compute the "best" route. The main considerations for successful packet forwarding include the following:

- Path length
 - Path length is the sum of the costs associated with each link. A commonly used value for this metric is called *hop count*, the number of routing devices, i.e. routers/firewalls, through the path that a packet takes to travel from the source to its destination.

- Bandwidth
 - Bandwidth is the traffic capacity of a path, rated by "Mbps".
- Load
 - Load refers to the usage of a router. The usage can be evaluated by CPU utilization and the throughput.
- Delay
 - Delay is regarding to the time it takes to move a packet from the source to the destination. The time depends on many factors, such as the bandwidth, load, and the length of the path.

Different routing protocols rely on one or several metrics to examine and evaluate the links in the network. Regarding the goals of the design, the algorithm uses its metrics to decide the optimal paths.

10.3.3 OSPF

OSPF is the embedded dynamic routing algorithm in D-Link firewalls. From the previous section, we see the main characteristics of OSPF as a Link state routing algorithm. Now we look at the actual operation of this algorithm.

Areas & Routers

OSPF features hierarchical routing to give better support to complex network environment. Since today's network is getting more and more sophisticated, the size of the whole routing table, time required for routing computation, and the traffic for information exchange for a large network become excessive. OSPF enables the administrator to partition the AS(autonomous system) into several smaller areas, so that the burden for routing computation and routes maintenance on each router is reduced.

An OSPF area is a group of computer hosts and routers within an AS, identified by a unique **area ID**, and every OSPF router has a unique **router ID** with the format of an IP address.

On top of the OSPF hierarchy is a central area called **backbone area** to which all the other areas in the AS must connect. The backbone area is responsible for distributing routing information among all the connecting areas and has the area ID 0.0.0.0. In some cases where it is not possible to physically connect to the backbone area, a **Virtual Link (VLink)** can be

configured to connect to the backbone through a non-backbone area. VLink can also be used to link through partitioned backbone areas.

A normal OSPF area acts like a private network connecting to the backbone area via some router called **Area Border Router(ABR)**. ABRs have interfaces in more than one area, and maintains separate routing information databases for each area to which they are connecting by an interface. The routers reside in the same OSPF area only need to learn and synchronize link-state information with the ABR.

Some Routers that exchange routing information with routers in other ASs are called **Autonomous System Boundary Routers(ASBRs)**. ASBRs introduce externally learned routes to the AS and flood the external routing advertisement throughout all OSPF normal areas.

To reduce the flooding traffic of external routes advertisement, a special area called **"stub area"** can be configured. When a stub area is configured, the ABR will automatically advertise a default route so that routers in the stub area can reach destinations outside the area. The default route becomes the single exit point of the stub area, and the area will not accept broadcasts of external routes.

Operating Process

Establishment – "Hello"

At the initialization stage, each router within an area detects its directly connected network, and sends "Hello" packets to all its OSPF enabled interfaces to determine who their neighboring routers are. Routers having interfaces directly connected and residing in the same OSPF area become neighbors.

When a router sends and receives "Hello" packets and detects multiple routers in an AS, it will select a **Designated Router(DR)** and also a **Backup Designated Router(BDR)** for further link-state information exchange.

DR and BDR are automatically elected by "Hello" protocol on every OSPF broadcast network. The **Router Priority** which is configurable on a per-interface basis is the parameter that controls the election. The router with the highest priority number becomes

DR and the next highest one becomes BDR. If priority number 0 is specified for a router, it will not be eligible in the DR/BDR election.

Once the DR and BDR are selected, all the other routers within the same OSPF area establish a relationship with them to further exchange routing information. Any router switched on later will accept the existing DR/BDR on the network regardless of its own router priority. Since there is high demand on DR's central link-state information control, **Router Priority** should be configured to elect the most reliable router on a network as DR. BDR is elected at the same time as DR, and has the same relationship establishment with other routers in the area, in order to make the transition to a new DR smoother if there is any failure of the primary DR.

In addition to relationship establishment, "Hello" packets also act as **Keepalive** messages to keep track of the reachability of the links. The "Hello" packets are sent periodically with a predefined interval to let routers know that other routers are still functional.

Update – "LSA"

Each router maintains "state" and "link" information for routing. A "link" is an interface on the router and the "state" of the link is the information including the interface address, network, neighbor routers, and so on. These "*link-state*" information is stored in a router's data structure called "*link-state database*".

When a router determines the DR by the "Hello" packet, it will generate a *link-state advertisement (LSA)* and send it to DR. The DR controls and updates its central link-state database and distribute the database information to all the other routers in the same OSPF area.

After the initial exchange and the building of the database, each router within the same OSPF area will contain an identical database, which is a complete topology map of the area including the cost of the links. Due to any change in routing information, a router will save a new copy of link state into its database and send LSA to DR. The DR then flood the update to all participating routers in the area to synchronize the link-state database.

Path determination – "SPF"

After the database of each router is fully exchanged and synchronized, the router will calculate a *Shortest Path First (SPF) tree* to all known destinations based on the database. By running the SPF algorithm, each router will be able to determine the best (least-cost) path for data forwarding to any destination in the area. The destination, associated cost, and the next hop to reach the destination will be appended into each router's IP routing table.

Upon any update to the link-state database, a router will recalculate the shortest path tree and update the routing table.

OSPF Authentication

Authentication is available as an optional securing method for the OSPF environment. A router can validate the identity of another router during the link-state information exchange. OSPF authentication can be either *none*, *simple*, or *MD5*. With simple authentication, the passphrase goes in clear-text over the link, while with MD5 message-digest algorithm, the key will not pass over the link directly. Thus, MD5 should be considered as a more secure authentication mode. More information about encryption, message digest, and authentication can be found in [20.2 Introduction to Cryptography](#).

10.4 Route Failover

The route failover feature can be used when there is two or more routes to a destination. For instance in a scenario where two ISP:s are available to connect to the Internet. One ISP, the primary, is used in the normal case, and a backup ISP is used when the primary ISP is down.

Routes can be monitored in two ways. A monitored route can be considered down if link status on the interface is down, or if the default gateway doesn't answer on ARP requests. It is possible to use both monitoring methods at the same time.

10.4.1 Scenario: Route Failover Configuration



Example: Two ISPs

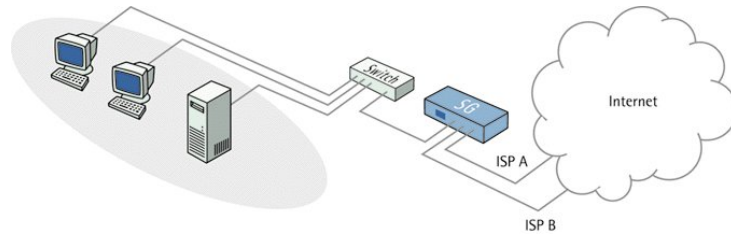


Figure 10.1: Route Failover Scenario

In this scenario shown in figure 10.1, two ISP:s (ISP A and ISP B) are used to connect to the Internet. ISP A is connected to the WAN1 interface of the firewall and ISP B is connected to interface WAN2. In order to configure the D-Link firewall to use ISP A as primary ISP, and ISP B as backup ISP, monitored routes will have to be configured.

We will need two routes, one default route (0.0.0.0/0) with metric 1, that use the default gateway of ISP A and one default route with metric 2 that use the default gateway of ISP B.

WebUI :

1. Turn off auto configuration of routes.

First of all we need to make sure that no routes are automatically added for interface WAN1 and WAN2.

Interfaces → Ethernet → Edit (WAN1):

On the "Advanced" tab, enter the following:

Add default route if default gateway is specified: Disable

Then click **OK**

Interfaces → Ethernet → Edit (WAN2):

On the "Advanced" tab, enter the following:

Add default route if default gateway is specified: Disable

Then click **OK**

2. Add default route over the WAN1 interface.

Next step is to add default route for interface WAN1.

Routes → **Main Routing Table** → **Add** → **Route**:

Enter the following:

General

Interface: WAN1

Network: 0.0.0.0/0

Gateway: Default gateway of ISP A.

Local IP Address: (None)

Metric: 1

Monitor

Monitor This Route: Enable

Monitor Interface Link Status: Enable

Monitor Gateway Using ARP Lookup: Enable

Then click **OK**



- It is possible to manually configure the ARP lookup interval to use. The chosen value should be at least 100 ms. If multiple routes are monitored on the same interface, a higher value may have to be chosen to make sure that the network is not flooded with ARP requests. ■

3. Add default route over the WAN2 interface.

Next step is to add default route for interface WAN2.

Routes → **Main Routing Table** → **Add** → **Route**:

Enter the following:

General

Interface: WAN2

Network: 0.0.0.0/0

Gateway: Default gateway of ISP B.

Local IP Address: (None)

Metric: 2

Then click **OK**

4. Create interface group and add rules.

To be able to write rules with destination interface that can use either route, you have to create a interface group that use the Security/Transport Equivalent feature. This makes the two interfaces equal in a security sense.

Creating the interface group.

Interfaces → **Interface Groups** → **Add** → **Interface Group**:

Enter the following:

Name: Type a name for the interface group.

Security/Transport Equivalent: Enable

Interfaces: Select the WAN1 and WAN2 interface.

Then click **OK**

Add rules.

Add rules using the newly created interface group as destination interface. See [14.3 IP Rules Configuration](#) for details on how to configure rules.



- The default route for interface WAN2 will not be monitored. The reason for this is that we have no backup route for the route over interface WAN2.
-

10.5 Dynamic Routing Implementation

In D-Link firewalls, the implementation of dynamic routing involves two primary configuration tasks: *OSPF process* & *dynamic routing policy*.

10.5.1 OSPF Process

OSPF process configured in the firewall groups OSPF participating firewalls and routers into OSPF areas. Each process enabled on a router is given a unique *Router ID* in an IP address format and an *authentication method* is chosen.

The areas are defined on the basis of the firewall's interfaces. An interface that belongs to an area has a *Routing Priority* to be used for the area's *DR election*. The interface can either be used for *broadcast*, *point-to-point*, or *point-to-multipoint* communication. The broadcast interface learns neighboring routers automatically by flooding "Hello" packets, while for point-to-point or point-to-multipoint interface, one or more specific neighbors need to be configured for the interface manually. *Routing metrics* used for OSPF can also be set or modified on an interface to interfere in the OSPF path determination.

Once the OSPF process is properly configured for the firewall, it can begin to talk with other firewalls/routers using OSPF algorithm, and learn the link-state information of the network.

10.5.2 Dynamic Routing Policy

Based on the routing information learned by the OSPF process, dynamic routing policy forms a filter to the information and tells the firewall what to do with those knowledge by defined actions.

A *Dynamic Routing Policy rule* filters statically configured or OSPF learned routes according to parameters like the origin of the routes, destination, metric, and etc. The matched routes can be controlled by the *actions* to be either exported to OSPF processes or to be added to one or more routing tables.


The most common usages of Dynamic Routing Policy are listed as follows, examples are given next.

- Importing OSPF routes from OSPF process into the routing table.

- Exporting routes from the routing table to OSPF process.
- Exporting routes from one OSPF process to another OSPF process.

10.5.3 *Scenarios: Dynamic Routing Configuration*

In this section, basic scenarios for using OSPF Process and Dynamic Routing Policy are illustrated.

 **Example:** Setting up OSPF process

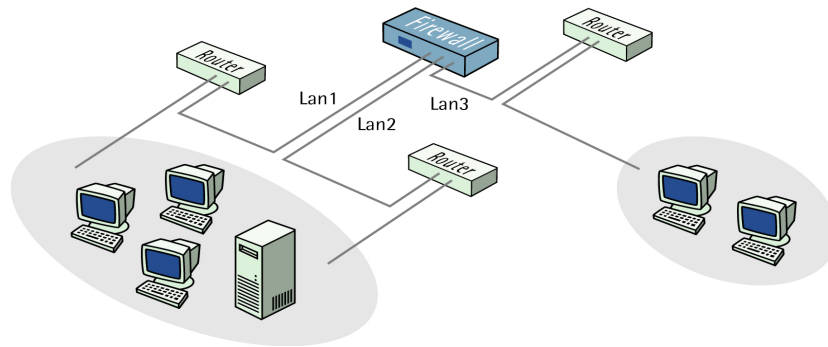


Figure 10.2: OSPF Process Scenario

As shown in Figure 10.2 , the firewall is assumed to have an interface "lan3" connected to a couple of local networks, which the firewall will control the only path into; and 2 interfaces, "lan1" and "lan2" attached to a larger local network. Some of the networks will be accessible through both interfaces, so that some redundancy might be achieved if one path becomes unreachable. This is done by placing the two interfaces "lan1" and "lan2" into a **security equivalent interface group**.

An OSPF process named as "ospf-proc1" is created, and only one OSPF area, the backbone "area0" (0.0.0.0), is used in this example. The 3 involved interfaces are added into the area to make the firewall participate in the OSPF process. However, this will not add any learned routes into any routing table, nor will it inform its neighbor about any static routes in its routing table(s)(except the routes for the 3 interfaces participating in this OSPF process). To control this information exchange, dynamic routing

policy need to come into play (See the next 2 scenarios for dynamic routing policy).

WebUI :

1. OSPF Process:

– adding an OSPF process called "ospf-proc1".

Routing → OSPF Processes → Add → OSPF Process: → General:

Name: ospf-proc1

→ **Authentication:**

Select one of the authentication types to be used in the process (none, password, or MD5).

Then click **OK**

2. Area:

– specifying an area to the "ospf-proc1" process.

In the "ospf-proc1" configuration page:

Add → Area:

General:

Name: "area0"

Area ID: 0.0.0.0

Then click **OK**.

3. Interfaces:

– adding the participating interfaces into the process.

In the "area0" configuration page:

Interfaces → **Add** → **Interface**:

→ **General**:

Interface: select "lan1" from the dropdown list.

("lan1" is assumed to have been defined in Ethernet interfaces)

Interface Type: select "Auto"

Metric/Bandwidth:

Either set a metric value or specify a bandwidth, e.g. 100Mbit.

→ **Advanced**:

Make sure that the configuration values listed here correspond with the values used by the other OSPF neighbors in the network.

Then click **OK**.

Repeat this step to add interfaces "lan2" and "lan3".

4. Interface Group:

– placing "lan1" and "lan2" in a security equivalent interface group.

Interfaces → **Interface Groups** → **Add** → **Interface Group**:

General:

Name: select a name for the group, e.g. "ifgrp-ospf1".

Check **Security/Transport Equivalent** check box.


Interfaces:

Select "lan1" and "lan2" from **Available** list and put them into the **Selected** list.

Then click **OK**.



- Make sure that the firewall's IP rules, which allowing traffics going through these interfaces, use this interface group as source interface. ■

 **Example:** Importing routes from an OSPF AS into the main routing table

It is assumed that a previously configured OSPF process named "ospf-proc1" has been created.

In this scenario, all received routes from "ospf-proc1" will be added into the main routing table, as this is not done automatically in the D-Link firewall.

WebUI :

1. Dynamic Routing Rule:

Routing → **Dynamic Routing Policy** → **Add** → **Dynamic Routing Rule**:

General

Name: e.g. "importOSPFRoutes".

Check **From OSPF Process**:

Select "ospf-proc1" from **Available** list and put it into **Selected** list.

Destination Network

...**Or is within:** all-nets

Then click **OK**.

2. Routing Action:

In the "importOSPFRoutes" configuration page:

→ **Routing Action** → **Add** → **Add Route**:

General

Destination:

Select the main routing table from **Available** list and put it into **Selected** list.

Then click **OK**.

The result of this configuration is that all learned routing information will be added to the main routing table as long as they don't override any static

routes or previously inserted default routes.



Example: Exporting the default route into an OSPF AS

It is assumed that a previously configured OSPF process named "ospf-proc1" has been created.

In this scenario the default route (only) from the main routing table will be exported into the OSPF process "ospf-proc1".

WebUI :

1. Dynamic Routing Rule:

Routing → Dynamic Routing Policy → Add → Dynamic Routing Rule:

General

Name: e.g. "exportDefRoute"

Check From Routing Table:

Select the main routing table from **Available** list and put it into **Selected** list.

Destination Network

Exactly Matches: all-nets

Then click **OK**.

2. OSPF Actions:

In the "exportDefRoute" configuration page:


→ OSPF Actions → Add → Export OSPF:

General

Export to process: Select "ospf-proc1" from the dropdown list.

Then click **OK**.

10.6 Scenario: Static Routing Configuration

 Example: Creating a Static Route

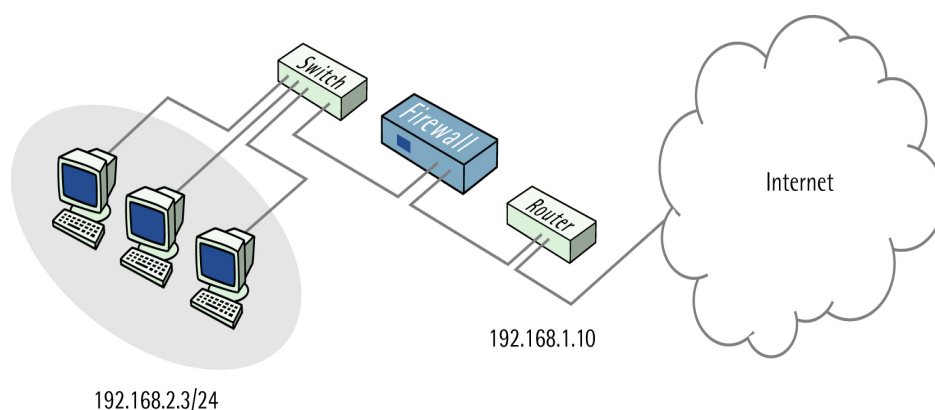


Figure 10.3: Static Routing Scenario

In this example a network 192.168.2.0/24 has been setup to be routable through a router(192.168.1.10) on the local network, as shown in Figure 10.3. To enable the firewall to communicate with that network (through interface "lan"), a static route must be configured.

WebUI :

Routing → Main Routing Table → Add → Route:

General:

Interface: Select "lan".

Network: Select the network address object (192.168.2.0/24).

Gateway: Select the router's address object (192.168.1.10).

Then click **OK**.

This will allow the firewall to route traffic destined for the 192.168.2.0/24 network through the router at 192.168.1.10.



- As a result of this setup the return traffic from the router will be routed directly upon the local network with a standard "Allow" rule set. For this scenario to work the IP rule set must either dictate that the traffic for this network is to be NATed or forwarded without state tracking (FwdFast). ■

10.7 Policy Based Routing(PBR)

10.7.1 Overview

Policy Based Routing(PBR) is an extension to normal routing described previously, which offers network administrators significant flexibility to implement their own defined policies on making routing decisions. By PBR, packets can go through a user desired route other than the routing algorithms decided one.

Normal routing forwards packets according to destination IP address information derived from static routes or dynamic routing protocol. For example, by OSPF, the router will only take the least-cost(shortest) path that obtained from SPF calculation to transport packets. Complementing to this destination-address-solely concern, PBR gives more control over routing by enabling the router to use specific path for certain traffic flow based on various criterion, such as *source addresses* and *service types*.

Moreover, D-Link firewalls extend the benefits of PBR further by not just looking at the packets one by one, but also at state information, so that the policy can provide control on both forward and return directions.

PBR can be applied to applications including:

- Source sensitive routing
 - When more than one ISP is used to provide Internet services, PBR can route traffic originating from different sets of users through different paths across the firewall.
- Service based routing
 - PBR can route certain protocols through transparent proxies, such as Web caches and anti-virus scanners.

- Creating provider-independent metropolitan area networks
 - All users share a common active backbone, but can use different ISPs, subscribing to different streaming media providers.

PBR implementation in D-Link firewalls consists of two elements:

- One or more named PBR tables in addition to the normal routing table.
- A separate PBR ruleset, which determines which named routing table to use.

10.7.2 Policy-based Routing Tables

Policy-based routing tables are alternative tables additional to the main routing table. These tables contain the same fields for describing routes as the main routing table, except that there is an **Ordering** parameter defined on each of them. This parameter dictates when the PBR table comes into play in firewall's route lookup, either prior or later than the main table.

10.7.3 Policy-based Routing Policy

The rules defined in PBR policy are selectors of different routing tables. Each PBR rule is triggered by the fields of service type and source & destination interface and network. During the firewall's lookup, the first matching rule is carried out, and routes can be chosen and prioritized by the order parameter on a per-state basis other than packet-by-packet lookup, which means that PBR rules can specify which routing table to use in both *forward* and *return* direction.

10.7.4 PBR Execution

The sequence of PBR execution cooperating with the main routing table and the firewall's rules setting can be summarized as follows:

1. Main routing table checking – looking up the interface for the packets' destination address.
2. Rules consulting – looking up in the firewall's *Rules* list to determine the action to the packets.
3. PBR policy consulting – If the lookup in step 2 results in allowing the packets to go through, the firewall will perform a lookup in the PBR

rules. The first matching rule will be the one to use. According to the specification in the rule, a routing table is selected to use. If there is no matching rule, the PBR tables will not be used and nor PBR will be performed. The firewall will forward the packets according to the main routing table only.

4. Address translation – If NAT rule was encountered in the rules consulting in step 2, address translation will be performed.
5. Final route lookup and packet forwarding – the firewall makes the final route lookup in the routing table decided in step 3, and forward the packet.

The decision of which routing table to use is made before carrying out address translation. However, the actual route lookup is performed on the altered address.



Example: Creating a Policy-Based Routing Table

In this example we create a policy-based routing table named "TestPBRTTable".

WebUI :

Create PBR Table

Routing → **Policy-based Routing Tables** → **Add**
→ **Policy-based Routing Table:**

Name: TestPBRTTable

Ordering:

First - means that the named routing table is consulted first of all. If this lookup fails, the lookup will continue in the main routing table.

Default - means that the main routing table will be consulted first. If the only match is the default route (0.0.0.0/0), the named routing table will be consulted. If the lookup in the named routing table fails, the lookup as a whole is considered to be failed.

Only - means that the named routing table is the only one consulted. If this lookup fails, the lookup will not continue in the main routing table.

Remove Interface IP Routes: If enabled, the default interface routes are removed, i.e. routes to the *core* interface, which are routes to the firewall itself.

Then click **OK**

**Example:** Creating a Policy-Based Route

After defining the PBR table "TestPBRTTable", we add routes into the table in this example.

WebUI :

Create PBR Route

Routing → **Policy-based Routing Tables** → **TestPBRTTable** → **Add** → **Route**:

Enter the following:

Interface: The interface to route over.

Network: The network to route.

Gateway: The gateway to send routed packets to.

Local IP Address: The IP address specified here will be automatically published on the corresponding interface. This address will also be used as the sender address in ARP queries. If no address is specified, the firewall's interface IP address will be used.

Metric: Specifies the metric for this route. (Mostly used in route fail-over scenarios)

Then click **OK**

10.7.5 Scenario: PBR Configuration

The following example illustrates a multiple ISP scenario which is a common use of policy based routing.

**Example:** Multiple ISPs

This scenario assumes the following:

- Each ISP will give you an IP network from its network range. We will assume a 2-ISP scenario, with the network 1.2.3.0/24 belonging to "ISP A" and "2.3.4.0/24" belonging to "ISP B". The ISP gateways are 1.2.3.1 and 2.3.4.1, respectively.
- All addresses in this scenario are public addresses, for simplicity's sake.

- This is a "drop-in" design, where there are no explicit routing subnets between the ISP gateways and the firewall.

In a provider-independent metropolitan area network, clients will likely have a single IP address, belonging to either one or the other ISP. In a single-organization scenario, publicly accessible servers will be configured with two separate IP addresses: one from each ISP. However, this difference does not matter for the policy routing setup itself.

Note that, for a single organization, Internet connectivity through multiple ISPs is normally best done through BGP, where you do not need to worry about different IP spans or policy routing. Unfortunately, this is not always possible, and this is where policy based routing becomes a necessity.

We will set up the main routing table to use ISP A, and add a named routing table, "r2" that uses the default gateway of ISP B.

Contents of the main routing table:

Interface	Network	Gateway	ProxyARP
LAN1	1.2.3.0/24		WAN1
LAN1	2.3.4.0/24		WAN1
WAN1	1.2.3.1/32		LAN1
WAN2	2.3.4.1/32		LAN1
WAN1	0.0.0.0/0	1.2.3.1	

Contents of the named policy routing table r2:

Interface	Network	Gateway
WAN2	0.0.0.0/0	2.3.4.1

The table r2 has its Ordering parameter set to Default, which means that it will only be consulted if the main routing table lookup matches the default route (0.0.0.0/0).

Contents of the Policy-based Routing Policy:

Source Interface	Source Range	Destination Interface	Destination Range	Service	Forward PBR	Return PBR
LAN1	2.3.4.0/24	WAN2	0.0.0.0/0	ALL	r2	<main>
WAN2	0.0.0.0/0	LAN1	2.3.4.0/24	ALL	<main>	r2



- We add rules for inbound as well as outbound connections. ■

Complete the following steps to configure this example scenario in the firewall.

1. Add routes to main routing table.

Add the routes found in the list of routes in the main routing table, as shown earlier.

See section [10.6 Creating a Static Route](#) for more information on how to add routes.

2. Create PBR table.

See section [10.7.4 Creating a Policy-Based Routing Table](#) for more information on how to create a PBR table. Name the table "r2" and make sure the ordering is set to "Default".

3. Add default route to PBR table.

Add the route found in the list of routes in the named policy routing table "r2", as shown earlier.

See section [10.7.4 Creating a Policy-Based Route](#) for more information on how to add routes to a PBR table.

4. Add PBR policies.

We need to add two PBR policies according to the list of policies shown earlier.

Routing → Policy-based Routing Policy → Add → Policy-based Routing Rule:

Enter the information found in the list of policies displayed earlier.

Repeat this step to add the second rule.

10.8 Proxy ARP

Proxy ARP is the process in which one system responds to the ARP request for another system. For example, host A sends an ARP request to resolve the IP address of host B. Instead of Host B, the firewall responds to this ARP request.

In essence, Proxy ARP has the same functionality as ARP publish (See [9.6 ARP](#))

The biggest difference here is that you can, in a simple manner, publish entire networks on one or more interfaces at the same time. Another, slightly less significant difference is that the firewall always publishes the addresses as belonging to the firewall itself; it is therefore not possible to publish addresses belonging to other hardware addresses.



- It is only possible to Proxy ARP on a Ethernet and VLAN interfaces. ■

CHAPTER 11

Date & Time

Correctly set date and time is of greatest importance for the product to operate properly. For instance, time scheduled policies and auto-update of IDS signatures are two features that require the clock to be correctly set. In addition, log messages are tagged with time stamps in order to point out exactly when a specific event did occur. Not only does this assume a working clock, but also that the clock is being synchronized with other devices in the network.

To maintain current date and time, the product makes use of a built-in real-time clock. The clock is also equipped with a backup battery to ensure operation even if the product should have lost its power. In addition, the product supports time synchronization protocols in order to automatically adjust the clock based on information from other devices.

11.1 Setting the Date and Time

11.1.1 Current Date and Time



Example:

To adjust the current date and time, follow the steps outlined below:

WebUI :

System → **Date and Time:**

General

Click the **Set Date and Time** button

In the pop-up window,

Date: select the current year, month and day in the dropdown lists.

Time: enter the current hours, minutes, and seconds in the edit box.

Then click **OK**.



- The new current date and time will be applied instantly. ■

11.1.2 Time Zone

The time zone setting should be set to reflect the time zone where the product is physically located.



Example:

To modify the time zone, follow the steps outlined below:

WebUI :

System → **Date and Time:**

Time zone and daylight saving time settings

Time zone: select the appropriate time zone in the dropdown list.

Then click **OK**.

11.1.3 Daylight Saving Time(DST)

Many regions honor *Daylight Saving Time (DST)* (or summer time as it is called in many countries). Daylight saving time works by advancing the clock during summer to get more out of the summer days. Unfortunately, the principles regulating daylight saving time vary in different countries, and in some cases there are even variants within the same country. For this reason, the product does not automatically know when to adjust for DST. Instead, this information has to be manually provided if daylight saving time is to be used.

There are basically two parameters governing daylight saving time; the DST period and the DST offset. The DST period specifies on what dates daylight saving time starts and ends, respectively. The DST offset indicates the number of minutes to advance the clock during the daylight saving time period.



Example:

To enable DST, follow the steps outlined below:

WebUI :

System → **Date and Time:**

Time zone and daylight saving time settings

Check **Enable daylight saving time**

Offset: enter the number of minutes the clock should be advanced during DST.

Start Date: select the starting date for DST period in the dropdown list.

End Date: select the ending date.

Then click **OK**.

11.2 Time Synchronization

The clock in the product is likely to be fast or slow after a period of operation. This is normal behavior in most network and computer equipment and is commonly solved by utilizing a time synchronization mechanism.

The product is able to adjust the clock automatically based on information received from one or several timeservers in the network. Using time synchronization is highly recommended, as it ensures the product to have its date and time aligned with other products in the network, or even with public timeservers providing highly accurate time information based on atomic clocks.

11.2.1 Time Synchronization Protocols

The product supports two kinds of protocols to be used for time synchronization:

- **SNTP**
 - Defined by RFC 2030, The *Simple Network Time Protocol (SNTP)* is a lightweight implementation of the *Network Time Protocol (NTP)* described in RFC 1305.
- **UDP/TIME**
 - The *Time Protocol (UDP/TIME)* is an older method of providing time synchronization service over the Internet. The protocol provides a site-independent, machine-readable date and time. The time service sends back to the originating source the time in seconds since midnight on January first 1900.

Most current public timeservers are using the NTP protocol.

11.2.2 Timeservers

Up to three timeservers can be configured to query for time information. By using more than one single server, situations where an unreachable server causes the time synchronization process to fail can be prevented. Please note that the product always queries all configured timeservers in order to compute an average time based on the responses from all servers. Search engines on the Internet can be used to find updated lists of publicly available timeservers.

11.2.3 Maximum Adjustment

To avoid situations where a faulty timeserver causes the product to update its clock with highly erroneous time data, a maximum adjustment value (in seconds) can be specified. If the difference between the current time in the product and the time received from a timeserver is greater than the maximum adjustment value, that timeserver response will be discarded. For example, assume that the maximum adjustment value is set to 60 seconds, and that the current time in the product is 16:42:35. If a timeserver responds with a time of 16:43:38, the difference is 63 seconds, which is not acceptable according to the maximum adjustment. Thus, no update will occur for that response. The default maximum adjustment value is 36,000 seconds.

11.2.4 Synchronization Interval

The interval between each synchronization attempt can be adjusted if needed. By default, this value is 86,400 seconds (1 day), meaning that the time synchronization process is executed one time per day.



Example: Enabling Time Synchronization using SNTP

In this example, time synchronization is being setup using the SNTP protocol and using NTP servers installed at the Swedish national laboratory for time and frequency.

WebUI :

System → **Date and Time:**

Automatic time synchronization

Check **Enable time synchronization**.

Select the following from the dropdown lists:

Time Server Type: SNTP

Primary Time Server: dns:ntp1.sp.se

Secondary Time Server: dns:ntp2.sp.se

Tertiary Time Server: (None)

Click **OK**.

■  **Note** ■

- This example uses domain names instead of IP addresses. Therefore, make sure the **DNS client settings** of the system are properly configured as described in [12 DNS](#). ■

CHAPTER 12

DNS

Domain Name System (DNS) can be considered as a gigantic distributed database that is used to translate from computer names to their IP addresses.

DNS is used within the firewall whenever there is need to translate a domain name to an IP address. Also, the DHCP server within the firewall can hand out the DNS servers configured in the firewall to all clients that request an IP lease. The example below describes how to configure DNS servers in D-Link firewalls. The configured servers are used by the internal DNS client as well as other subsystems such as the DHCP server.



Example: Configuring DNS server(s)

WebUI :

System → **DNS**:

Primary Server: Enter the IP address of the primary DNS server or select the address object from the dropdown list (if the address of the server has been defined in **Address Book**).

Secondary Server: (Optional)

Tertiary Server: (Optional)

Then click **OK**.

CHAPTER 13

Log Settings

In the Administration part, we have introduced the general concepts of logging and the design in D-Link firewalls to cope with significant events (refer to [5, Logging](#)). In this chapter, we present configuration examples for enabling logging function.

Except for some default logging events that will be generated automatically, for example, the firewall's startup and shutdown, logging needs to be enabled manually in specific sections of the firewall's configuration.

To set up logging in D-Link firewalls, the following two steps are required:

1. Define one or several log receivers.
2. Enable the logging function in certain sections.

13.1 Implementation

13.1.1 Defining Syslog Receiver

As explained in [Section 5.2.1](#), Syslog receivers are external log managers used for receiving and storing log data generated by the firewall. Log data are sent to the Syslog receiver(s) through messages, which are defined by *Facility* and *Severity*.

Facility defines how messages are sent to a Syslog receiver by specifying source identifiers in the messages. The receiver can typically sort messages

from different sources based on the identifier. The valid range is 0 to 7, representing Syslog facilities "local0" through "local7".

Severity is the degree of emergency attached to the logged event message for debug. D-Link firewalls can be set to send messages at different severity levels. Sorted from highest to lowest importance, these levels are : *Emergency, Alert, Critical, Error, Warning, Notice, Info, and Debug.*



Example: Defining a Syslog receiver

To define a Syslog receiver in the firewall, follow the steps below:

WebUI :

System → **Log and Event Receivers** → **Add** → **Syslog Receiver:**
General

Name: Enter a name for the receiver.

IP Address: Select the address object from the dropdown list (if the address of the receiver has been defined in **Address Book**), or enter the IP address directly into the edit box.

Facility: Choose one of the facilities from the dropdown list.

Port: 514 (by default)

Then click **OK**.

13.1.2 Enabling Logging

After setting up one or more receivers, logging need to be enabled to function. In WebUI, all configuration items that can generate log events have a page named **Log Settings** in their properties window. This page contains options to enable logging, and to specify certain log receiver(s) and severity for the event.



Example: Enabling Syslog

In this example, we assume an IP rule has been defined previously, and enable logging on this rule to monitor its action to traffics.

WebUI :

Rules → IP Rules: Click the IP rule item → **Log Settings:**

General

Check **Enable logging**

Severity: Choose one of the severity levels from the dropdown list.

Log Receivers

Log to: Check either **All receivers** or **Specific receiver(s)**

(If **Specific receiver(s)** is checked, select the Syslog receiver(s) that has been defined previously from **Available** list to **Selected** list.)

Then click **OK**.



Example: Enabling Memory Log

The firewall's built-in memory log receiver can be enabled in a similar way as explained in the Example: Enabling Syslog.

To check the log file contents stored by the memory log receiver, follow the steps below:

WebUI :

Menu Bar: Status → Logging:

100 items of newly generated events can be displayed per page.

To see previous events, press **next**.

Part VI

Security Polices

Security policies regulate the manner of network applications to protect from abuse and inappropriate use. D-Link firewalls feature for providing various mechanisms to aid the administrators in building security polices for attacks prevention, privacy protection, identification, and access control.

Topics in this part includes:

- [IP Rules](#)
- [Access \(Anti-spoofing\)](#)
- [DMZ & Port Forwarding](#)
- [User Authentication](#)

CHAPTER 14

IP Rules

14.1 Overview

The list of rules defined on the basis of network objects – addresses, protocols, services – is the heart of any firewall. Rules determine the basic filtering functions of the firewall, which is essential. Following the rules configuration, the firewall regulates what is allowed or disallowed to go through, and how address translation, bandwidth management, and traffic shaping, is applied to the traffic flow. Any ambiguous or faulty rule may loose the security control or make the firewall useless.

Basically, there are two stances of the firewall that describe fundamental philosophy of security:

- ◇ The *default deny* stance:
Everything is denied unless specifically permitted. ✓
- ◇ The *default permit* stance:
Everything is permitted unless specifically denied.

In order to provide the highest possible level of security, *default deny* is the default policy in D-Link firewalls. The default deny is accomplished without a visible rule in the list. However, for logging purposes, rule list commonly has a *DropAll* rule at the bottom with logging enabled.

When a new connection is being established through the firewall, the list of rules are evaluated, top to bottom, until a rule that matches the new

connection is found. The action of the rule is then carried out. If the action is *Allow*, the connection will be established and a **state** representing the connection is added to the firewall's internal state table. If the action is *Drop*, the new connection will be refused.

First matching principle – If there are several matching rules, the first matching one decides what will happen to the connection. (Except for SAT rules, shown in [Example](#).)

Consecutive packets belonging to an existing connection will not need to be evaluated again. Instead, a highly optimized **state-lookup** algorithm will search the internal state table for an existing state representing the connection. This methodology is applied not only on TCP connections, but on UDP and ICMP traffic as well. Thus, the size of the firewall ruleset does not affect the throughput of the firewall.

A rule is expressed in a definite form, consisting of two logical parts: *the fields* and *the action*. The subsections below explain the parameters of a rule that are available in D-Link firewalls.

14.1.1 Fields

Fields are some pre-defined and reusable network objects, such as *Addresses* and *Services*, which are used by every rule for matching purpose. The following fields in the rule list are used by the firewall to check a packet in the traffic flow. All these filtering fields have to match the contents of a packet for any rule to trigger.

- **Service:** the protocol type that the packet must match.
(Services are defined as logical objects before configuring the rules, see [8.2 Services](#))
- **Source Interface:** one or a group of interfaces where the packet is received on the firewall.
- **Source Network:** the network that the source IP address of the packet matches.
- **Destination Interface:** one or a group of interfaces where the packet is aiming at.
- **Destination Network:** the network that the destination IP address of the packet matches.

14.1.2 Action types

When all the fields listed in the previous section are matched by a packet, a rule is triggered, and certain action specified by the matching rule will be carried out. The types of actions include:

- **Allow:**
Lets the packet pass through the firewall. The firewall will also set up a 'state' to remember the connection, and pass the rest of the packets in this connection through its stateful inspection engine.
- **NAT:**
Works like *Allow* rules, but with dynamic address translation enabled. (See [14.2.2 NAT](#))
- **FwdFast:**
Lets the packet pass through the firewall *without* setting up a state for it. Generally speaking, it is faster for an individual packet, but it is less secure than *Allow* or *NAT* rules, and also slower than *Allow* rules for the whole established connection, as every subsequent packet also needs to be checked against the rule section.
- **SAT:**
Tells the firewall to perform static address translation. (See [14.2.3 Static Address Translation](#)) This rule also requires a matching *Allow*, *NAT* or *FwdFast* rule further down. (See [Example](#))
- **Drop:**
Tells the firewall to immediately discard the packet.
- **Reject:**
Acts like *Drop*, but will return a TCP-RST or ICMP-Unreachable message, telling the sender that the packet was disallowed.

14.2 Address Translation

14.2.1 Overview

For functionality and security considerations, *Network Address translation(NAT)* is widely applied for home and office use today. D-Link firewall provides options to support both *Dynamic* and *Static* NAT. These two types are represented by the NAT and SAT rule settings respectively.

This section explains how NAT works and what it can and cannot do.

14.2.2 NAT

What is NAT?

When communicating on the Internet, each node needs to register a unique network address to be reachable. But the available unique addresses from the range of IPv4 is very limited while nowadays network is becoming larger and larger. Network address translation (NAT) enables computers on private networks to use a set of unregistered addresses internally, and share one or a set of public IP addresses for external connections to Internet resources. Normally, a router or a firewall located at where the LAN meets the Internet makes all necessary IP address translations.

For each NATed network, the private IP address spaces (10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16) are reused. This means that multiple interfaces connected to different networks can have the same address, relieving the pressure of having to use public IPv4 addresses for every node.

Why is NAT widely used?

In addition to solve the IP shortage problem, NAT is developed to serve many other purposes:

- **Functionality** – Using NAT, there is no need to register an IP address for every computer in a local network. A company can use many internal IP addresses and one registered public IP address to provide Internet services. Since these addresses are used internally only, there is no possibility of address collision with other companies. It allows a company to combine multiple access connections into a single Internet connection.
- **Security** – Computers locating at the local network and using a range of private addresses are not directly accessible from the Internet. To

the outside world, the whole private network is like one node using one public IP address, and the inside structure and addresses of the network is hidden. NAT depends on a machine on the local network to initiate any connection to hosts on the other side of the firewall or the router, it prevents malicious activity initiated by outside hosts from reaching those local hosts. NAT-enabled firewalls, for example, D-Link firewalls, handle all the translation and redirection work for passing traffic and can provide ways to restrict access to the Internet at the same time.

- Flexibility of administration – NAT can be used to divide a large network into several smaller ones. The smaller parts expose only one IP address to the outside, which means that computers can be added or removed without impacting external networks. D-Link firewalls contain DHCP server, which allow clients to be configured automatically. The administrator does not need to apply any change to every computer in the internal network, for instance, if the DNS server address changes, all clients will automatically start using the new address the next time they contact the DHCP server.

How NAT works

In TCP/IP network communication, each IP packet contains a header with the source and destination addresses and port numbers (Source address: source port — Destination address: destination port). This combination completely defines a single connection. The addresses specify the two end computers of the link, and the two port numbers guarantee that every connection that belongs to a certain service is uniquely identified. Each connection is originated from a unique source port number in one end, and all reply packets from the other end contain the same number as their destination port, so that the initiator can relate them back to its correct connection.

A NAT-enabled firewall must change the source address on every outgoing packet to be its single public address. It therefore also renumbers the source port number to be unique, so that it can keep track of each connection. The firewall uses a mapping table to relate the real local address and source port plus its translated source port number to a destination address and port. When it receives any returning packets, it can therefore reverse the translation to route them back to the correct clients.

Because the mapping table relates complete connection information -

source and destination address and port numbers - it is possible to validate any or all of this information before passing the traffic. This checking helps the firewall to protect a private LAN against attacks from the outside.

NAT mechanism discard all traffic that does not match a mapping table entry, therefore it is also regarded as a security device. However, NAT is not a substitute for firewall rules. There are TCP and UDP ports open corresponding to applications and services running on the NAT. If the NAT device is a computer, rather than a dedicated firewall, then the computer is vulnerable to attack. Therefore, the recommendation is to use NAT-enabled firewall with rule settings specified for traffic.

14.2.3 Address translation in D-Link Firewall

D-Link firewalls support two types of address translation: *dynamic* (NAT hide), and *static* (SAT).

Dynamic Network Address Translation

The process of dynamic address translation involves the translation of multiple sender addresses into one or more sender addresses, like private IP addresses are mapped to a set of public IP addresses.



Example: Dynamic NAT

	Sender	\rightleftharpoons	Server
	192.168.1.5 : 1038	\rightarrow	195.55.66.77 : 80
FW_ tran	195.11.22.33: 32789	\rightarrow	195.55.66.77 : 80
reply	195.11.22.33: 32789	\leftarrow	195.55.66.77 : 80
FW_ rest	192.168.1.5 : 1038	\leftarrow	195.55.66.77 : 80

Table 14.1: Dynamic NAT.

Table 14.1 shows a example of dynamic NAT, The sender, e.g. 192.168.1.5, sends a packet from a dynamically assigned port, for instance, port 1038, to a server, e.g. 195.55.66.77 port 80.

Usually, the firewall translates the sender address to the address of the interface closest to the destination address. In this example, we use 195.11.22.33 as the public address. In addition, the firewall changes the

source port to a free port, usually one above 32768, 32789 is used here. The packet is then sent to its destination.

The recipient server regards the firewall NATed address as the origin of the packet, processes the packet, and sends its response back to the NATed address.

The firewall receives the packet and compares it to its list of open connections. Once it finds the connection in question, it restores the original address and forwards the packet to the real sender.

Static Address Translation (SAT)

SAT is a type of address translation in which a public IP address is statically mapped to a private IP address. Dynamic NAT is normally used for outgoing traffic, while SAT is used for incoming traffic. For example, using SAT allows an internal host, such as a Web server, to have an unregistered (private) IP address and still be reachable over the Internet. The private IP address of the server is mapped to a public static IP address, which can be seen from the Internet.

In D-Link firewalls, SAT is implemented to provide many important functions, for example:

- *DMZ & Port Forwarding*: SAT supports the use of DMZ network to provide public services to the Internet, meanwhile protecting the private network from unnecessary disclosure to the outside world. (see [16](#), *DMZ & Port Forwarding*)
- *Server Load Balancing*: SAT can redirect connections pointed at some server to randomly selected servers. (see [24](#), *Server Load Balancing*)

14.3 *Scenarios: IP Rules Configuration*

This section shows you example configurations of IP rules, including:

- NAT rule
- SAT rule
 - Publicly accessible Server with a Private Address in a DMZ

Other features' setups cooperating with NAT can be found in [16 DMZ & Port Forwarding](#), and [24 Server Load Balancing](#).



Example: Enabling Ping on firewall's external IP address

In this example, we configure an IP rule to allow ICMP(Ping) packets to be received by the external interface of the firewall.

1. Define an ICMP service object and name it "ping-inbound". (Note that the D-Link Firewall is delivered with a "ping-inbound" service configured as default which can be used)
2. Create a new Rule with name "Ping_to_Ext", and *Allow* the service from *Any* interface on *all-nets* to the firewall's *core* interface on *ip_ext* network.

WebUI :

1. Create Ping-Inbound Service

If no ping-inbound service is defined, we need to create a new service.

Objects → **Services** → **Add** → **ICMP Service:**

Name: ping-inbound

ICMP Parameters

ICMP Message Types: Echo Request (Codes 0-255)

Then click **OK**

2. Create Rule

Final step is to create the rule that will allow ICMP(Ping) packets to be received by the external interface of the firewall.

Rules → **IP Rules** → **Add** → **IP Rule**:

Name: Ping_to_Ext

Action: Allow

Service: ping-inbound

Source Interface: any

Source Network: all-nets

Destination Interface: core

Destination Network: ip_ext

Then click **OK**



Example: NAT rule

In this case, we set up a NAT rule in the firewall that will allow us to browse the Internet from private IP addresses behind the firewall. The private IP addresses will be translated to the external IP address of the firewall.

1. Add a "HTTP" service object that use TCP port 80.
2. Add a "DNS" service object that use TCP/UDP port 53 to enable name resolving service.
3. Create two rules that *NAT* the services above from the internal interface on the internal network to any destination interface on any network.

WebUI :

1. Create HTTP Service

If no http service is defined, we need to create a new service.

Objects → **Services** → **Add** → **TCP/UDP Service**:

Name: http

Type: TCP

Source: 0-65535

Destination: 80

Then click **OK**

2. Create DNS-All Service

If no dns-all service is defined, we need to create a new service.

Objects → **Services** → **Add** → **TCP/UDP Service**:

Name: dns-all

Type: TCPUDP

Source: 0-65535

Destination: 53

Then click **OK**

3. Create HTTP Rule

Next step is to create the rule that will NAT HTTP traffic from internal interfaces out on external interfaces.

Rules → **IP Rules** → **Add** → **IP Rule**:

Name: HTTP_from_LAN

Action: NAT

Service: http

Source Interface: LAN

Source Network: lan-net

Destination Interface: any

Destination Network: all-nets

Then click **OK**

4. Create DNS Rule

Final step is to create the rule that will NAT DNS traffic from internal interfaces out on external interfaces.

Rules → **IP Rules** → **Add** → **IP Rule**:

Name: DNS_from_LAN

Action: NAT

Service: dns-all

Source Interface: LAN

Source Network: lan-net

Destination Interface: any

Destination Network: all-nets

Then click **OK**

■  **Note** ■

- For NAT rules it is possible to specify the IP address that internal IP addresses should be translated to. This can be done on the "NAT" tab when configuring the rule. As default, the address of the destination interface is used. ■



Example: SAT rule

Publicly Accessible Server with a Private Address in a DMZ

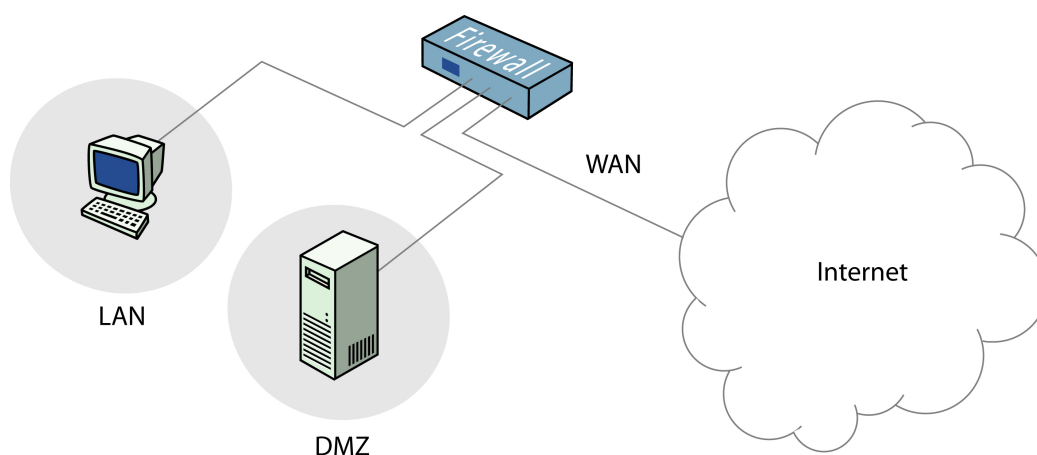


Figure 14.1: SAT Example.

This example features a web server with a private address located in a DMZ, and internal machines located on a local network that want to browse the Internet. In order to enable external users to access the web server, the server must be reachable from a public address. Thus, we translate port 80 on the firewalls external address to port 80 on the web server:

1. Add a "HTTP" service object that use TCP port 80.

2. Rule 1: Create a new rule that *SAT* HTTP traffic directed to the external public IP address *ip_ext*, to the private IP address of the web server.
 - * Rule 2: Create an *NAT* rule to permit traffic *SAT*:ed by the above rule.
 - * Rule 3: Create a *NAT* rule that allows internal machines on the local network to access the Internet via HTTP.

WebUI :

1. Create HTTP Service

If no http service is defined, we need to create a new service.

Objects → **Services** → **Add** → **TCP/UDP Service**:

Name: http

Type: TCP

Source: 0-65535

Destination: 80

Then click **OK**

2. Create HTTP SAT Rule

Next step is to create the rule that will *SAT* HTTP traffic directed to the external public IP address *ip_ext*, to the private IP address of the web server.

Rules → **IP Rules** → **Add** → **IP Rule**:

Name: SAT_to_WebServer

Action: *SAT*

Service: http

Source Interface: any

Source Network: all-nets

Destination Interface: core

Destination Network: ip_ext

SAT

Translate the: Destination IP Address

To New IP Address: ip_webserver

Then click **OK**

3. Create HTTP SAT/NAT Rule

Next step is to create an *NAT* rule to permit traffic SAT:ed by the above rule.

Rules → **IP Rules** → **Add** → **IP Rule:**

Name: SATNAT_to_WebServer

Action: NAT

Service: http

Source Interface: any

Source Network: all-nets

Destination Interface: core

Destination Network: ip_ext

Then click **OK**

4. Create HTTP NAT Rule

Final step is to create a *NAT* rule that allows internal machines on the local network to access the Internet via HTTP.

Rules → **IP Rules** → **Add** → **IP Rule:**

Name: HTTP_from_LAN

Action: NAT

Service: http

Source Interface: LAN

Source Network: lan-net

Destination Interface: any

Destination Network: all-nets

Then click **OK**



■

SAT needs a second rule

SAT rule needs a second rule line to pass traffic through (shown as the "Allow" rule above). The second rule can be a *Allow*, *FwdFast*, or *NAT*, and this second rule line must be placed below the initiating SAT rule.

The initiating SAT rule *does nothing* to the actual data. If there is a match with the packet received and a SAT rule, the firewall will

continue to pass the packets through the rule list until a second rule matches. When the packets are leaving the rule list, this rule redirects them to the destination.

Problem with the current rule set

This rule set makes the internal addresses visible to machines in the DMZ (see 16, *DMZ & Port Forwarding*). When internal machines connect to the firewall's external interface *ip_ext*, they will be allowed to proceed by Rule 2 without NAT (the first matching principle). From security perspective, all machines in the DMZ that provide public services should be regarded as any other Internet servers connected to untrusted networks.

Alternative Solutions

1. Keep Rule 1 and reverse the sequence of Rule 2 and Rule 3, so that the NAT rule is carried out for internal traffic before the *Allow* rule matches.
2. Keep Rule 1 and Rule 3, change Rule 2 so that it only applies to external traffic (most likely traffic from interface WAN) – an "Allow" rule to permit Rule 1 from *external* connections (most likely interface WAN) on *all-nets* to the firewalls external public address *ip_ext*.

■

**Tip**

Determining the best course of action and the sequential order of the rules must be done on a case-by-case basis, taking all circumstances into account.

CHAPTER 15

Access (Anti-spoofing)

15.1 Overview

The primary function of any firewall is to control the access to protected data resources, so that only authorized connections are allowed. Access control is basically addressed in the firewall's IP rules (introduced in [14. IP Rules](#)). According to the rules, the firewall considers a range of protected LAN addresses as trusted hosts, and restricts the traffic flow from the untrusted Internet going into the trusted area, and also the other way around.

One underlying flaw of this *trust* based control is that, it tends to neglect the potential hazard caused by masquerade. The clever attackers make tricks to fool the firewall by pretending the identity of a trust host, which is the so called *Spoofing* technique.

15.1.1 IP Spoofing

IP spoofing is one of the most common masquerading attacks, where the attacker uses IP addresses trusted by the firewall to bypass the traffic filtering. In the spoofing process, the header of an IP indicating the source address of a given packet can be easily modified to a local host's address, so that the firewall will believe the request came from a trusted source. Although the packet cannot be responded to the initial source, there is potential for unnecessary network congestion and denial of service (DoS)

attacks. Even if the firewall is able to detect the DoS attacks, it is hard to trace or stop it because of the spoofing.

15.1.2 Anti-spoofing

To equip the firewalls with *Anti-spoofing* capability, an extra filter against the source address verification is in need. D-Link firewalls provide the network administrators choices to do the source based IP filtering by *Access Rule*.

Other features provided by D-Link firewalls, such as *User Authentication* and *Encryption*, ensure that proper authentication measures are in place and communication are carried out over secure channels, which can also reduce the spoofing threats. (See [17 User Authentication](#), [VIII VPN](#))

15.2 Access Rule

15.2.1 Function

The *Access rule* is capable of monitoring traffic to verify that packets arriving on an interface of the firewall do not have a source address which is associated with a network of another interface. In other words, the principle of the rules can be described as follows:

- Any incoming traffic with a source IP address belonging to a local trusted host is NOT allowed.
- Any outgoing traffic with a source IP address belonging to an outside untrusted network is NOT allowed.

The first one prevents an outsider to use a local host's address as source address, and the second one prevents any local host to launch the spoofing.

The Access rule set act as an add on filter to the firewall's rules list, and ensures that the source addresses of packets received on a specific interface are always within the correct network, provided that the Access rule is correctly configured. If the Access section lookup does not produce a hit, the firewall will perform a reverse lookup in its routing table.

15.2.2 Settings

The configuration of an access rule is similar to normal rule, containing *Filtering Fields* and the *Action* to take. If the traffic matches all the fields,

the rule is triggered, and the specified action will be carried out by the firewall.

Filtering Fields

- **Interface:**
The interface that the packet arrives on.
- **Network:**
The IP span that the sender address should belong to.

Action

- **Drop:**
to *discard* the packets that match the defined fields.
- **Accept:**
to *accept* the packets that match the defined fields for further inspection in the Rule set.
- **Expect:**
If the sender address of the packet matches the *Network* specified by this rule, the receiving interface is compared to the specified interface. If the interface matches, the packet is accepted in the same manner as by the *Accept* action. If the interfaces do not match, the packet is dropped in the same manner as by the *Drop* action.

Logging can be enabled on demand for these Actions.
(Refer to [5 Logging](#))

15.3 Scenario: Setting up Access Rule



Example: Allowing Traffic from a Specific Network

This example will show how to make sure that traffic received on the *LAN* interface always have the correct source address, within the *lan-net* network.

WebUI :

1. Create Access Rule

The following rule will make sure that no traffic with a source address not within the *lan-net* network is received on the *LAN* interface.

Rules → **Access** → **Add** → **Access Rule:**

Name: LAN_Access

Action: Expect

Interface: LAN

Network: lan-net

Then click **OK**

CHAPTER 16

DMZ & Port Forwarding

16.1 General


16.1.1 Concepts

DMZ – "Demilitarized Zone" – stands for an area that is neither part of the trusted internal network nor directly part of the public Internet.

Typically, DMZ is a separate subnetwork between the firewall protected private LAN and the public network. It contains one or more computers that accessible to Internet traffic and acts like proxy servers for public services, such as Web(HTTP), FTP, SMTP(Email), and DNS servers.

In a DMZ configuration, the computers(servers) sitting outside the private LAN, respond to or forward the service requests. The firewall is configured to prevent computers in the DMZ from initiating inbound requests, and it forwards traffic from the Internet to DMZ computers without direct contact with the inner LAN. Obviously, this approach adds an extra layer of protection to the Intranet–firewall–Internet infrastructure.

D-Link firewalls offer supports to DMZ planning and protection through network object *Interface* and *Rules* configurations.

 **Example:** A corporation's Web server

We take a look at a simple example, showing one utilization of DMZ with a D-Link firewall.

The most common publicly available service that every corporation need to have is *Web browsing(HTTP)*. However, it is unsafe to place a Web server inside the internal network together with other private computers, because such server can easily be exploited in a harmful way by intruders. When the server falls into the control of a wrong hand, other private computers will be vulnerable to attacks. Therefore, such service should be located in a separate network area – DMZ.

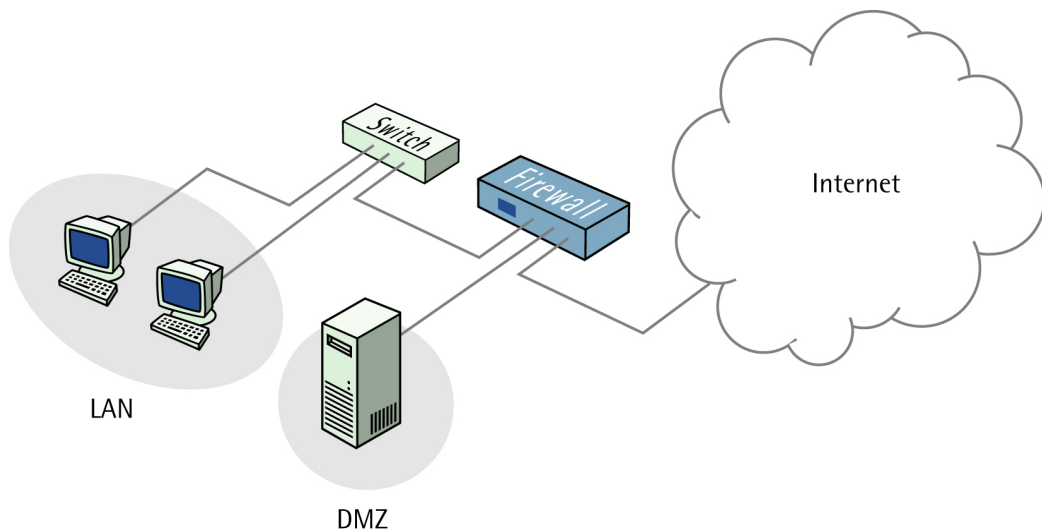


Figure 16.1: A Web Server in DMZ

In this example, we have a D-Link firewall connecting a private LAN, a DMZ subnetwork, and the Internet, shown in Figure 16.1. The firewall takes charge of a) all the connections from the Internet to the DMZ; b) necessary connections from the DMZ to the private LAN. The Web server is placed in the DMZ. Requests to Web browsing service go through the firewall, and are forwarded to the Web server.

We can define *Rules* that let the server in the DMZ accepts only certain types of service requests, *HTTP-based* requests in this case, to protect the

server. For instance, suppose our web server is running on NT that might be vulnerable to a number of denial-of-service attacks against services, such as RPC, NetBIOS and SMB. These services are not required for the operation of HTTP. So we can set rules to block relevant TCP connections to ports 135, 137, 138, and 139 on that server to reduce the exposure to denial-of-service attacks.

Summary:

This solution means that, with a DMZ deployment, there is no direct access from the Internet into the internal network, and anyone trying to access resources in DMZ from the Internet would have to pass the firewall's rules. The setting of the firewall's rules follows one important *security principle*, that is, limiting the connections to the *minimum* necessary numbers to support the services.

16.1.2 DMZ Planning

The utilization of DMZ is a large-scale work, involving segmentation of the network structure and firewall rule configurations. Therefore, it requires careful planning to achieve the protection and scalability purposes.

We use a small set of components to illustrate the different approaches of DMZ planning:

- A ***D-Link firewall*** with 3 interfaces: *Int net*, *DMZ net*, and *Ext net*
- A private computer: ***Client A***
- A ***File Server*** containing the LAN's private data
- A ***Database Server*** containing resources for public web services.
- A ***Web Server*** for public connections.

Approach 1 – File Server, Database Server, and Client A on *Int net*; Web Server on *DMZ net*.

Drawback: The Web server on *DMZ net* needs to open some ports on *Int net* to access the Database Server. If the Web Server is taken over by intrusion, the Database Server and other components on *Int net* may expose to attacks.

Approach 2 – Move the Database Server out to the DMZ network.

Drawback: Although all the public accessible data are now on the DMZ network, the protection to the Database Server is weakened. If a hacker takes control over the Web Server, he or she can go straight into the Database.

Approach 3 – Split DMZ into different zones. ✓

Solution: The best approach for this scenario is dividing the *DMZ net* into different subnetworks according to different services and security levels of the components. We put the Database Server and the Web Server on separate interfaces of the firewall, and configure access rules for each interface. If the hacker gets control of the Web Server, he or she still has very limited access to the Database Server.

16.1.3 Benefits

As illustrated in the previous section, making good use of a DMZ network provides several advantages on both network security and management's perspectives:

- Splitting services up not only by hosts, but by networks limits the level of trust among network components. This approach can greatly reduce the likelihood of penetration on one component being used to break into the others.
- Dividing DMZ into different zones helps to restrict security policies upon components that having different functions and levels of security.
- The scalability of the network architecture is increased by placing components on different subnetworks.

CHAPTER 17

User Authentication

17.1 Authentication Overview

Before any user's service request is authorized according to the firewall's security policies, the firewall need to verify the *identity* of the user, to ensure that the corresponsive user is who she or he claims to be.

Authentication is the process to address such issue. It forms a filter at the forefront of the firewall's access control, packet filtering, and secure tunneling. In this chapter, we concern the validity of the user, in term of person; the same principles apply to devices in the network as well.

17.1.1 Authentication Methods

Generally, the authentication process prompts the user to show one's credential with great care that this secret is not possessed by anyone else. The solutions and enabling technologies can be categorized upon the basis of:

a) *Something the user is*

The unique attributes of the user that are different on every person – physiological characteristics – such as one's *fingerprint, retina, or voice*.

b) *Something the user has*

The key "tool" that a user possesses, such as a *Digital Certificates, a Passcard, or Public & Private Keys*.

c) Something the user knows

The secret information that only the involved user knows and keeps, such as the most commonly used *Password* or a *Shared secret* phrase.

The difficulty of using method a) is that it requires some special devices to scan and read the feature presented, which are relatively expensive.

Another risk that may cause this to fail is that the features are almost impossible to have substitutes; in case the user loses the feature by accident, nothing can be used for replacement.

Therefore, the more commonly used methods for network services are (b) and (c). There are also potential risks by using either of these methods, for example, the keys may be intercepted, the card can be stolen, people tend to use weak passwords that are easy to guess, and they may be bad on keeping any secret, and so on. Thus, these two approaches are often combined to have add one factors and security levels. For example, a passcard is often granted to a person with a password.

User authentication is frequently used in services, such as HTTP, FTP, and VPN. D-Link firewalls use *Username/Password* as primary authentication method, strengthened by encryption algorithms. The basic concepts of encryption is covered by [20.2 Introduction to Cryptography](#). More advanced and secure means of authentication, such as the *Public-private Key System*, *X.509 Certificate*, *IPsec& IKE*, *IKE XAuth*, and *ID List* are introduced in: [20.2.2 Authentication & Integrity](#), and [22 VPN Protocols & Tunnels](#).

17.1.2 Password Criterion

In the *Username/Password* coupling, the username(account name) as an *identifier* tells who you are, and the password severs as an *authenticator* to prove that this is true. To penetrate certain system and obtain the user or administrator's privileges, the password is often subject to attacks.

Attacks

There are mainly three different ways to attack a password:

- Guess:
Try possible cases. Passwords that are chosen from a dictionary, or user's personal information, such as *name*, *telephone number*, and *birth date* are vulnerable to this attack.
- Find:

Find the notes that recording the password, or ask the user directly. Many people tend to write the passwords down on paper, and they may tell a password to someone they trust, which are potential leakages.

- **Crack:**
Exhaustive search by some software crackers. Passwords with short length or less random characters are easily cracked.

Counter Measures

To prevent the "Guess" and "Crack" attacks, a *GOOD* password should be:

- containing more than 8 characters with no repeating
- random characters which are not commonly used phrases
- containing small and capital characters
- containing numbers and special characters

For password maintenance, some guidelines are available as listed below:

- password should not be recorded anywhere – on paper or in a computer file
- never reveal your password to anyone
- passwords should be regularly changed to counter any undetected compromises
- choosing the passwords that can be typed fast to prevent observing by someone nearby.

Although the above conditions may seem strict and inconvenient, they are intended for securing both the users' rights and properties, and the protected network system. A good selection of password also helps in protecting the *Layer 2 tunnels*, which apply encryption on the basis of user input passwords (See [22.2 PPTP/L2TP](#)).

17.1.3 User Types

D-Link firewalls and authentication schemes give support to diverse users. The user types can be:

- administrators

- normal users accessing the network
- PPPoE/PPTP/L2TP users
 - using PPP authentication methods
- IPsec & IKE users
 - the entities authentication during the IKE negotiation phases (Implemented by *Pre-shared Keys* or *Certificates*. Refer to [22.1.4 IKE Integrity & Authentication.](#))
- IKE XAuth users
 - extension to IKE authentication, occurring between negotiation phase 1 and phase 2
- user groups
 - group of users that are subject to same regulation criterion

17.2 Authentication Components

D-Link firewalls can either use a locally stored database, or a database on an external server to provide user authentication.

17.2.1 Local User Database(UserDB)

The *Local User Database* is a built-in registry inside D-Link firewalls, containing the profiles of the legal users and user groups. Users' names and passwords can be configured into this database, and the users having same privileges can be grouped up to ease the administration.

One user can be stored as a member into more than one group, any change made to the group propagates to each group member. Passwords are stored in the configuration using *reversible cryptography*. This is in order to be compatible with various challenge-response authentication methods such as [CHAP](#), and so forth.

When the local user database is enabled, the firewall consults its internal user profiles to authenticate the user before approving any user's request.

17.2.2 External Authentication Server

In a larger network topology, it is preferable to have one central database within a dedicated server or a cluster of servers to handle all the

authentication information. When there are more than one firewall in the network and thousands of users added or removed constantly, the administrator will not have to configure and maintain separate databases of authorized user profiles on each firewall. Instead, the external server can validate the username/password against its central database, which is easily administered. D-Link firewalls support the use of *RADIUS*(*Remote Authentication Dial-in User Service*) *Server* to offer external authentication feature.

RADIUS is currently the most prevalent standard for remote authentication. As the protocol defines, it uses PPP to transfer the username/password message between RADIUS client and the server, and hence, applies the same authentication schemes as PPP, like [PAP](#) and [CHAP](#). Originally developed for dial-up remote access, RADIUS is now supported by VPN, wireless access points, and other network access types.

A RADIUS client, i.e. D-Link firewall, sends user credentials and connection parameter information in the form of a RADIUS message to a RADIUS server. The RADIUS server maintains all the users and user groups profiles. It authenticates and authorizes the RADIUS client request, and sends back a RADIUS message response. RADIUS authentication messages are sent as UDP messages via UDP port 1812. One or more external servers can be defined in the firewall to improve the availability of the RADIUS system.

To provide security for RADIUS messages, a common *shared secret* is configured on both the RADIUS client and the server. The shared secret enables basic encryption of the user's password when the RADIUS message is transmitted from the RADIUS client to the server, and is commonly configured as a relatively long text string. It can contain up to 100 characters and is case sensitive.

17.2.3 Authentication Agents

Four different agents built in the firewall can be used to perform username/password authentication. They are:

- HTTP
 - Authentication via web browsing. Users surf on the firewall and login either through a HTML form or a 401 Authentication Required dialog.

- HTTPS
 - Authentication via secure web browsing. Similar to HTTP agent except that **Host & Root Certificates** are used to establish SSL connection to the firewall.
(refer to [22.3 SSL/TLS \(HTTPS\)](#))
- XAUTH
 - Authentication during IKE negotiation in IPsec VPN (if the IPsec tunnel has been configured to require XAUTH authentication).
(refer to [22.1.4 IKE XAuth](#))
- PPP
 - Authentication when PPTP/L2TP tunnels are set up (if the PPTP/L2TP tunnel has been configured to require user authentication).
(refer to [9.4.1 PPP](#), and [22.2 PPTP/ L2TP](#))

17.2.4 Authentication Rules

A user authentication rule specifies:

- From where (i.e. receiving interface, source network) users are allowed to authenticate to the firewall;
- Which agent will be used by the firewall to prompt users the authentication request.
- Where is the location of the database that the firewall consults to perform the authentication, either in a local registry or from the external server;
- Different timeout restrictions to logout the authenticated users automatically.



- When using XAUTH agent, there is no need to specify the receiving interface, or source network, as this information is not available at the XAUTH phase. For the same reason, only one XAUTH user authentication rule can be defined. XAUTH is only used to set up IPsec VPN tunnels. ■

17.3 Authentication Process

A D-Link firewall proceeds user authentication as follows:

- A user connects to the firewall to initiate authentication.
- The firewall receives user's request from its interface, and notes in the IP rule set that this traffic is allowed to reach its core authentication agent.
- According to the authentication agent specified in the authentication rule, the firewall prompts the user with authentication request.
- The user replies by entering one's identification information – username/password.
- The firewall validates the information w.r.t the authentication source specified in the authentication rule, either the local database or an external database in a RADIUS server will be taken.
- If a matching entry in the database is found, the firewall responses the user with approval message, otherwise rejection.
- The firewall then forwards the approved user's further service requests to their desired destinations, if the service is allowed by an IP rule explicitly and the user is a member of the user(s)/group(s) defined on the address object of that rule. Requests from those failed in the authentication step are discarded.
- After a certain time period, the authenticated user will be automatically logged out according to the timeout restrictions defined in the authentication rule.

17.4 *Scenarios: User Authentication Configuration*

In this section, guidelines and examples for authentication through HTTP/HTTPS agent are covered. For more examples about PPP and XAuth, please refer to [9.4.2](#), PPPoE Client Configuration, and [22](#), VPN Protocols & Tunnels, respectively.



Example: Configuring the local user database

In the example of authentication address object in [8.1 Address Book](#), a user group "users" is used to enable user authentication on "lanet". This example shows how to configure a user group in the firewall's built-in database.

WebUI :

1. **User Authentication** → **Local User Databases** → **Add** → **LocalUserDatabase:**

General

Enter a name for the new "lanet" user group folder:

Name: lanet_auth_users

Comments: folder for "lanet" authentication user group – "users"

2. **lanet_auth_users** → **Add** → **User:**

General

Username: Enter the user's account name here, e.g. "user1".

Password: Enter the user's password.

Confirm Password: Repeat the password above to avoid any mistyping.


Groups: One user can be specified into more than one group. Enter the group names here separated by comma, e.g. "users" for this example.

Then click **OK**.

3. Repeat step 2 to add all the "lanet" users having the membership of "users" group into the **lanet_auth_users** folder.

■  **Note** ■

- There are two default user groups, the administrators group and the auditors group. Users that are members of the administrators group are allowed to change the firewall configuration, while users that belong to the auditors group are only allowed to view the firewall configuration. Press the buttons under the **Groups** edit box to grant these group memberships to a user. ■

 **Example:** Configuring a RADIUS server

An external user authentication server can be configured by following the steps below:

WebUI :

User Authentication → **External User Databases** → **Add** → **External User Database**:

General

Name: Enter a name for the server here.

Type: The only type supported currently is **Radius**.

IP Address: Enter the IP address of the server here, or enter the symbolic name if the server's address has previously been defined in the **Address Book**.


Port: 1812 (RADIUS service uses registered UDP port 1812 by default.)

Retry Timeout: 2 (The firewall will resend authentication request to the sever if there is no response after the timeout, e.g. every 2 seconds. The firewall will retry three times as maximum.)

Shared Secret: Enter a text string here for basic encryption of the RADIUS messages.

Confirm Secret: Retype the string to confirm the one typed above.

and then click **OK**

 **Example:** Enabling HTTP authentication via local user database

To enable user authentication via a Web page, first, we need to add an **Allow** rule in the firewall's IP rules to let the firewall accept user's Web browsing to its HTTP(TCP port 80) agent; second, we specify a user authentication rule to tell the firewall how to perform the authentication, such as which database to take for user's profile lookup, and also the timeout restrictions; Third, another IP rule for dealing with service requests from authenticated users should be appended under the **Allow** rule from the first step. As explained in [14 IP Rules](#), all the other traffics that are not explicitly allowed by the IP rule, for example, the unauthenticated traffic coming from the interface where authentication is

defined, will be dropped by the firewall.

The configurations below shows how to enable HTTP user authentication to the user group "users" on "lanet". Only users that belong to the group "users" can have Web browsing service after authentication, as it is defined in the IP rule.

We assume that "lanet", "users", "lan_ip", local user database folder – "lanet_auth_users", and an authentication address object "lanet_users" have been specified (Refer to 8.1 Example: Enabling User Authentication for an IP Object).

WebUI :

1. Rules → IP Rules → Add → IP rule:

General

Name: http2fw

Action: Allow

Service: HTTP

(See 8.2.1 Example: Specifying a TCP service – HTTP)

Address Filter

Choose the following from the drop down lists:

Source

Destination

Interface: lan

core

Network: lanet

lan_ip

Comments:

Allow HTTP connections to the firewall's authentication agent.

Click **OK**.

2. **User Authentication** → **User Authentication Rules** → **Add** → **User Authentication Rule**

→ **General**

Name: HTTPLogin

Agent: HTTP

Authentication Source: Local

Interface: lan

Originator IP: lannet

Comments: HTTP authentication for lannet via local user database.

→ **Authentication Options**

General

Local User DB: lannet_auth_users

→ **HTTP(s) Agent Options**

General

Login Type: HTMLForm.

Click **OK**.

3. **Rules** → **IP Rules** → **Add** → **IP rule:**

General

Name: Allow_http_auth

Action: NAT

Service: HTTP

Address Filter

Source

Destination

Interface: lan

any

Network: lannet_users

all-nets

(Note here the source network is an address object containing user authentication information.)

Comments: Allow authenticated "users" from "lannet" to Web browsing onto Internet.

Click **OK**.



- 1. HTTP authentication will collide with WebUI's remote management service which also uses TCP port 80. To avoid this, please change WebUI port in **Advanced Settings** for **Remote Management** before proceeding the authentication configuration, for example, using port 81 instead.
 2. In **HTTP(s) Agent Options**, there are two login types available, **HTMLForm** and **BasicAuth**. The problem with **BasicAUTH** is that Web browsers cache the username and password entered in the 401- Authentication Required dialog. This is normally no problem if the browser is closed down, as it then clears the cache; but for systems with the browser imbedded in the operating system, the cache is harder to clear. Therefore, HTMLForm is recommended. A **Realm String** can be defined to be shown in the 401- Authentication Required dialog for BasicAUTH option.
 3. Timeout can be adjusted in **User Authentication** → **User Authentication Rules** → **Restrictions**. The options are **Idle Timeout** and **Session Timeout**.
 - **Idle Timeout:** If a user has successfully been authenticated, and no traffic has been seen from his IP address for this number of seconds, he/she will automatically be logged out. The value is 1800 by default.
 - **Session Timeout:** If a user has successfully been authenticated, he/she will automatically be logged out after this many seconds, regardless of if the firewall has seen activity from the user or not.
 - **Use timeouts received from the authentication server** checkbox: Some RADIUS servers can be configured to return idle-timeout and session values. If this checkbox is selected, the firewall will try to use these timeouts, prior to the timeout values specified above. If no timeouts are received from the authentication server, the timeout values specified above will be used.
 4. Another **Restrictions** configuration is the **Multiple Username Logins**. Three options are available as explained below:

- **Allow multiple logins per username**– If this one is selected, the firewall will allow users from different source IP addresses, but with the same username, to be simultaneous logged on.
- **Allow one login per username, disallow the rest**– If this is selected, the firewall will only allow one simultaneous username to be logged on. That is, if a user from another IP address tries to authenticate with the same username as that of an already authenticated user, the firewall will disallow this login.
- **Allow one login per username**– If this is selected, the firewall will only allow one simultaneous username to be logged on. If a user from another IP address tries to authenticate with the same username as that of an already authenticated user, the firewall will check if the authenticated user has been idle for a period of time. If so, the old user will be removed, and this new user will be logged in. If not, the new login-request will be rejected. The time period for this option can be defined in the edit box.

■

Part VII

Content Inspection

In addition to inspect the packets at the network layer (OSI layer 3), D-Link firewalls are capable of examining the content of each packet to give far more powerful and flexible protection on higher layers.

Topics in this part includes:

- [Application Layer Gateway \(ALG\)](#)
- [Intrusion Detection System \(IDS\)](#)

CHAPTER 18

Application Layer Gateway (ALG)

18.1 Overview

To complement the limitations of packet filtering, which only inspect in the packet headers, such as IP, TCP, UDP, and ICMP headers, D-Link firewalls embed an *Application Layer Gateway (ALG)* to support higher level protocols that have address information within the payload.

The ALG acts as the user's representative to obtain most commonly used Internet applications outside the protected network, e.g. Web access, file transfer, and multimedia. Such application-aware agents provide higher security than packet-filtering-only firewalls, since they are capable of scrutinizing all traffic for specific service protocols to give protection at the top level of the TCP/IP stack.

In this chapter, the following application standards supported by D-Link ALGs are described.

- [FTP](#)
- [HTTP](#)
- [H.323](#)

18.2 FTP

The File Transfer Protocol (FTP) is a TCP/IP-based protocol to exchange files between a client and a server. The client initiates the connection by connecting to the FTP server. Normally the client needs to authenticate itself by providing a predefined login and password. After granting access, the server will provide the client with a file/directory listing from which it can download/upload files (depending on access rights). The FTP ALG is used to manage FTP connections through the firewall.

18.2.1 FTP Connections

FTP uses two communication channels, one for control commands and one for the actual files being transferred.

When an FTP session is opened, the FTP client establishes a TCP connection (the control channel) to port 21 (by default) on the FTP server. What happens after this point depends on the mode of FTP being used.

Modes

There are two modes, *active* and *passive*, describing the role of server in respect to opening the data channels

In *active mode*, the FTP client sends a command to the FTP server indicating what IP address and port the server should connect to. The FTP server establishes the data channel back to the FTP client using the received address information.

In *passive mode*, the data channel is opened by the FTP client to the FTP server, just like the command channel. This is the *recommended* default mode for FTP clients, according to the "firewall-friendly FTP" RFC.

Security Issues

Both modes of FTP operation present problems for firewalls. Consider a scenario where an FTP client on the internal network connects through the firewall to an FTP server on the Internet. The IP rule in the firewall is then configured to allow network traffic from the FTP client to port 21 on the FTP server.

When *active mode* is used, the firewall is not aware of that the FTP server will establish a new connection *back to* the FTP client. Therefore, the connection for the data channel will be dropped by the firewall. As the port number used for the data channel is dynamic, the only way to solve this is to allow traffic from all ports on the FTP server to all ports on the FTP client. Obviously, this is not a good solution.

When passive mode is used, the firewall does not need to allow connections *from* the FTP server. On the other hand, the firewall still does not know what port the FTP client tries to use for the data channel. This means that the firewall has to allow traffic from all ports on the FTP client to all ports on the FTP server. Although this is not as insecure as in the active mode case, it still presents a potential security threat. Furthermore, not all FTP clients are capable of using passive mode.

Solution


The FTP ALG solves this problem by fully reassembling the TCP stream of the command channel and examining its contents. Thus, the firewall knows what port to be opened for the data channel. Moreover, the FTP ALG also provides functionality to filter out certain control commands and provide a basic buffer overrun protection.

The most important feature of the FTP ALG is its unique capability to perform on-the-fly conversion between active and passive mode. The conversion can be described like this:

- The *FTP client* can be configured to use passive mode, which is the recommended mode for clients.
- The *FTP server* can be configured to use active mode, which is the safer mode for servers.
- When a FTP session is established, the firewall will automatically and transparently receive the passive data channel from the FTP client and the active data channel from the server, and tie them together.

This implementation results in that both the FTP client and the FTP server work in their *most* secure mode. Naturally, the conversion also works the other way around, that is, with the FTP client using active mode and the FTP server using passive mode.

18.2.2 Scenarios: FTP ALG Configuration

 Example: Protecting a FTP Server

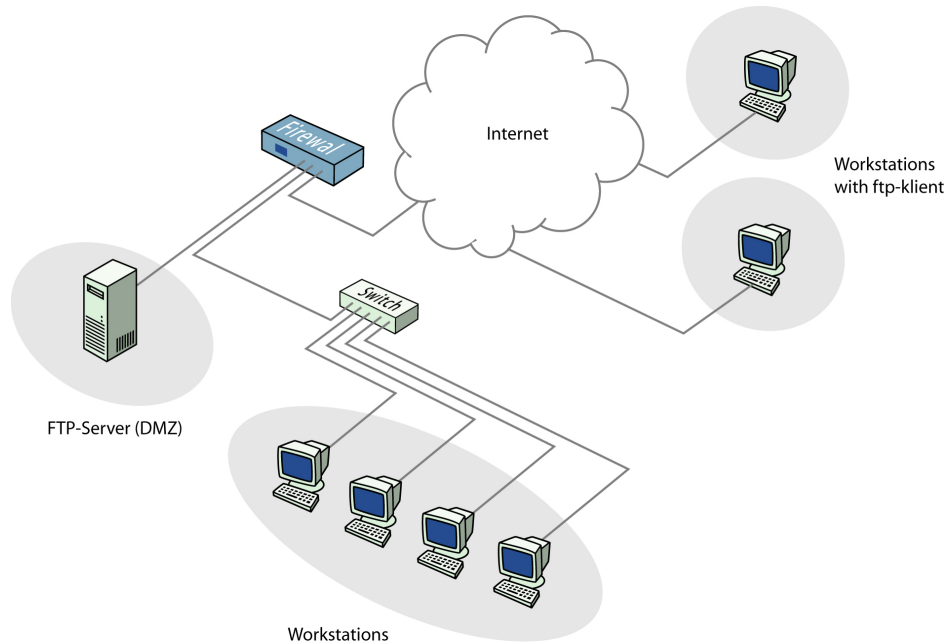


Figure 18.1: FTP ALG Scenario 1

In this example, a FTP Server is connected to a D-Link firewall on a DMZ with private IP addresses, shown in Figure 18.1. To make it possible to connect to this server from the Internet using the FTP ALG, the FTP ALG and firewall rules should be configured as follows:

WebUI :

1. ALG

Objects → **Application Layer Gateways** → **Add** → **FTP ALG**:

General:

Name: ftp-inbound

Check **Allow client to use active mode (unsafe for client)**.

Uncheck **Allow server to use passive mode (unsafe for server)**

Then click **OK**.

2. Services

Objects → **Services** → **Add** → **TCP/UDP Service**:

General:

Enter the following:

Name: ftp-inbound

Type: select TCP from the dropdown list.

Destination: 21 (the port the ftp server resides on).

Application Layer Gateway:

ALG: select "ftp-inbound" that has been created.

Then click **OK**.

3. Rules

– Allow connections to the public IP on port 21 and forward that to the internal FTP server:

Rules → **IP Rules** → **Add** → **IP Rule**:

General:

Name: SAT-ftp-inbound

Action: SAT

Service: ftp-inbound

Address Filter:

Source	Destination
--------	-------------

Interface: any	core
-----------------------	------

Network: all-nets	ip-ext
--------------------------	--------

(assume the external interface has been defined as "ip-ext")

SAT:

Check **Translate the Destination IP Address**

To:

New IP Address: ftp-internal.

(Assume this internal IP address of FTP server has been defined in the **Address Book** object.)

New Port: 21.

Then click **OK**.

– Traffic from the internal interface need to be NATed:

Rules → IP Rules → Add → IP Rule:

General:

Name: NAT-ftp

Action: NAT

Service: ftp-inbound

Address Filter:

Source	Destination
--------	-------------

Interface: dmz	core
-----------------------	------

Network: dmz-net	ip-ext
-------------------------	--------

NAT:

Check **Use Interface Address**

Then click **OK**.

– Allow incoming connections (SAT needs a second Allow rule):

Rules → IP Rules → Add → IP Rule:

General:

Name: Allow-ftp

Action: Allow

Service: ftp-inbound

Address Filter:

Source	Destination
--------	-------------

Interface: any	core
-----------------------	------

Network: all-nets	ip-ext
--------------------------	--------

Then click **OK**.



Example: Protecting FTP Clients

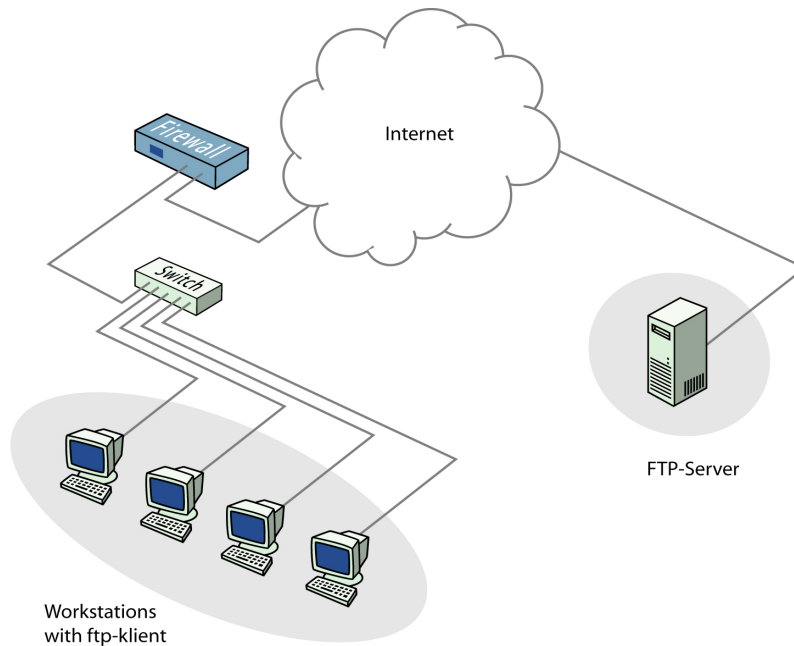


Figure 18.2: FTP ALG Scenario 2

In this scenario, shown in Figure 18.2, a D-Link firewall is protecting a workstation that will connect to FTP servers on the internet. To make it possible to connect to these servers from the internal network using the FTP ALG, the FTP ALG and firewall rules should be configured as follows:

WebUI :

1. ALG

Objects → **Application Layer Gateways** → **Add** → **FTP ALG**:

General:

Enter a descriptive name for the ALG.

Name: ftp-outbound

Uncheck **Allow client to use active mode (unsafe for client)**.

Check **Allow server to use passive mode (unsafe for server)**

Then click **OK**.

2. Services

Objects → **Services** → **Add** → **TCP/UDP Service**:

General:

Enter the following:

Name: ftp-outbound

Type: select TCP from the dropdown list.

Destination: 21 (the port the ftp server resides on).

Application Layer Gateway

ALG: select "ftp-outbound" that has been created.

Then click **OK**.

3. Rules (Using Public IPs)

The following rule need to be added to the IP rules in the firewall if the firewall is using **public** IP's; make sure there is not rules disallowing or allowing the same kind of ports/traffic before these rules. The service in use is the "ftp-outbound", which should be using the ALG definition "ftp-outbound" as described earlier.

– Allow connections to ftp-servers on the outside:

Rules → **IP Rules** → **Add** → **IP Rule**:

General:

Name: Allow-ftp-outbound

Action: Allow

Service: ftp-outbound

Address Filter:

Source	Destination
---------------	--------------------

Interface: lan	wan
-----------------------	-----

Network: lannet	all-nets
------------------------	----------

Then click **OK**.

4. Rules (Using Private IPs)

If the firewall is using **private** IP's, the following NAT rule need to be added instead.

Rules → **IP Rules** → **Add** → **IP Rule**:

General:

Name: NAT-ftp-outbound

Action: NAT

Service: ftp-outbound

Address Filter:

Source	Destination
---------------	--------------------

Interface: lan	wan
-----------------------	-----

Network: lannet	all-nets
------------------------	----------

NAT:

Check **Use Interface Address**

Then click **OK**.

18.3 HTTP

Hyper Text Transfer Protocol (HTTP), is the primary protocol used to access the *World Wide Web (WWW)*. It is a connectionless, stateless application layer protocol (OSI layer 7), which is based on the request/response architecture. The client, such as Web browser, typically sends a request by establishing a TCP/IP connection to a particular port (usually port 80) on a remote server. The server answers with a response string, followed by a message of its own, for example, a HTML file to be shown in the Web browser, an active-x component to be executed on the client, or an error message.

18.3.1 Components & Security Issues

To enable more advanced functions and extensions to HTTP services, some add-on components, known as "*active contents*", are usually accompanied with the HTTP response to the client computer.

ActiveX objects

An *ActiveX* object is a HTTP component, which is downloaded and executed on the client computer. Because it is executed on the client, certain security issues exist, which could cause harm to the local computer system.

JavaScript/VBScript

In order to display more advanced and dynamic HTML pages, scripts can be used. A script is executed by the web browser, and can be used to control the browser functionality, validate user's input, or a number of other features. It could potentially be used by an attacker in an attempt to cause harm to a computer system, or to cause various annoyances, such as pop-up windows.

Java Applets

A java applet is written in JAVA programming language, and a java-enabled browser can download and execute this code on the client computer. An applet can contain malicious code, which leads to security problems.

Cookies

A cookie is a small text file, stored locally on the client computer. Its objective is to make a web server remember certain information about a user, which has been entered previously. This can also contain confidential information.

18.3.2 Solution

D-Link firewalls address the security issues shown in the previous section by *Stripping Contents* and *URL Filtering*.

Stripping Contents

In D-Link HTTP ALG configuration, some or all of the active contents mentioned previously can be stripped away from HTTP traffic upon administrator's requests.

URL Filtering

A *Uniform Resource Locator (URL)* is an address to a resource on the WWW. This can for example be a HTML page, or a file resource. As a part of a security policy, it might be useful to restrict access to certain sites, or even to block certain file types to be downloaded. The opposite requirement could also be true – it might be favorable to allow full access (i.e. no removal of the above mentioned objects) to certain trusted resources.

A URL can be blacklisted in order to prevent access to it, while a whitelisted URL allows full access to the specific resource.



Example: Configuring HTTP ALG

In this example, a HTTP ALG in a D-Link firewall is created. It is configured to strip **ActiveX** objects, which will block displays such as Macromedia flash and shockwave. An undesired address is added into the blacklist. Restrictions to other active contents, or whitelists for trusted addresses can be configured in a similar way. We assume that the HTTP service object and an IP rule to allow the HTTP traffic have been defined in the firewall.

WebUI :

1. ALG

Objects → **Application Layer Gateways** → **Add**
→ **HTTP ALG**:

General:

Enter a descriptive name for the ALG.

Name: http-activex

Active Content Handling

Check **Strip ActiveX objects (including Flash)**

Then click **OK**.

After clicking ok, the configuration page goes to **URL Filtering** list.

Add → **HTTP URL**:

General

Action: select Blacklist from the dropdown list.

URL: Enter an undesired address in the edit box.

Then click **OK**.

2. Service

– Adding the HTTP ALG into the corresponding service object.

Objects → **Services** → **HTTP**:

Application Layer Gateway

ALG: select "http-activex" that has been created.

Then click **OK**.

18.4 H.323

18.4.1 H.323 Standard Overview

H.323 is a standard that is used for real-time audio, video and data communication over packet-based networks (e.g. IP). It specifies the components, protocols and procedures providing multimedia communication.

H.323 is a standard approved by the International Telecommunication Union to promote compatibility in video conference transmissions over IP networks.

H.323 is considered to be the standard for interoperability in audio, video and data transmissions as well as Internet phone and voice-over-IP (VoIP).

18.4.2 H.323 Components

The H.323 standard consists of these four main components:

- Terminals
- Gateways
- Gatekeepers

- MCUs (Multipoint Control Units)

Terminals

A H.323 terminal is a device that is used for audio and optionally video or data communication. For example phones, conferencing units, or software phones (for example: NetMeeting) running on standard PCs.

Gateways

A gateway connects two dissimilar networks and translates traffic between them. A H.323 gateway provides connectivity between H.323 networks and non-H.323 networks such as public switched telephone networks (PSTN). The gateway takes care of translating protocols and converting media between the different networks. A gateway is not required for communication between two H.323 terminals.

Gatekeepers

The Gatekeeper is a component in the H.323 system which is used for addressing, authorization and authentication of terminals and gateways. It can also take care of such things as bandwidth management, accounting, billing and charging. The gatekeeper may allow calls to be placed directly between endpoints, or it may route the call signaling through itself to perform functions such as follow-me/find-me, forward on busy, etc. A gatekeeper is needed when there is more than one H.323 terminal behind a NATing firewall with only one public IP.

MCUs (Multipoint Control Units)

MCUs provide support for conferences of three or more H.323 terminals. All H.323 terminals participating in the conference call have to establish a connection with the MCU. The MCU then manages the calls, resources, video and audio codecs used in the call.

18.4.3 H.323 Protocols

The different protocols used in H.323 is shortly described below:

H.225 RAS Signaling and Call Control (Setup) Signaling

The H.225 protocol is used for call signaling, that means that it's used to establish a connection between two H.323 endpoints (terminals). This call signal channel is opened between two H.323 endpoints or between a H.323 endpoint and a gatekeeper. For communication between two H.323 endpoints, TCP 1720 is used. When connecting to a gatekeeper, UDP port 1719 (H.225 RAS messages) are used.

H.245 Media Control and Transport

The H.245 protocol provides control of multimedia sessions established between two H.323 endpoints. The most important task for this protocol is to negotiate opening and closing of logical channels. A logical channel is, for instance, an audio channel used for voice communication. Video and T.120 channels are also called logical channels during negotiation.

T.120

The T.120 standard is made up of a suite of communication and application protocols. Depending on the type of H.323 product, the T.120 protocol can be used for application sharing, file transfer and conferencing features such as whiteboards.

18.4.4 H.323 ALG Overview

The H.323 ALG is a flexible application layer gateway that allows H.323 devices such as H.323 phones and applications to make and receive calls between each other while connected to private networks secured by D-Link Firewalls.

The H.323 specification was not designed to handle NAT, as IP addresses and ports are sent in the payload of H.323 messages. The H.323 ALG modifies and translates H.323 messages to make sure that H.323 messages will be routed to the correct destination and allowed through the firewall.

The H.323 ALG has the following features:

- H.323 version 5 (H.225.0 v5, H.245 v10)
- Application sharing (T.120)
- Gatekeeper support

- NAT, SAT

The H.323 ALG supports version 5 of the H.323 specification. This specification is built upon H.225.0 v5 and H.245 v10. In addition to support voice and video calls, the H.323 ALG supports application sharing over the T.120 protocol. T.120 uses TCP to transport data while voice and video is transported over UDP.

To support gatekeepers, the ALG makes sure to monitor RAS traffic between H.323 endpoints and the gatekeeper, in order to configure the firewall to let calls through.

NAT and SAT rules are supported, allowing clients and gatekeepers to use private IP addresses on a network behind the firewall.

18.4.5 *Scenarios: H.323 ALG Configuration*

The H.323 ALG can be configured to suit different usage scenarios.


It is possible to configure if TCP data channels should be allowed to traverse the firewall or not. TCP data channels are used by the T.120 protocol (see [18.4.3](#)), for instance. Also, the maximum number of TCP data channels can be limited to a fixed value.

The gatekeeper registration lifetime can be controlled by the firewall in order to force re-registration of clients within a time frame specified by the administrator.

Presented here are a few network scenarios, visualized in network diagrams. The scenarios are examples of network setups where the H.323 ALG is suitable to use. For each scenario a configuration example of both the ALG and the rules are presented.

The three service definitions used in these scenarios are:

- Gatekeeper (UDP ALL → 1719)
- H323 (H.323 ALG, TCP ALL → 1720)
- H323-Gatekeeper (H.323 ALG, UDP → 1719)

 Example: Protecting a Phone Behind a D-Link Firewall

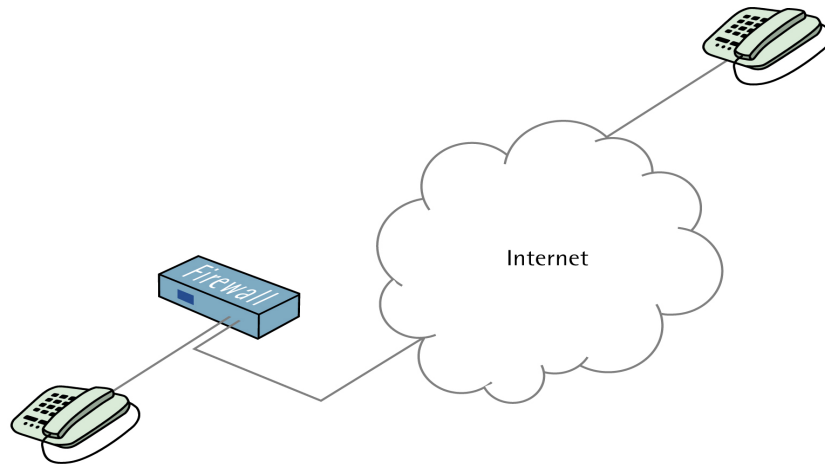


Figure 18.3: H.323 Scenario 1.

Using Public IP Addresses

In the first scenario a H.323 phone is connected to a D-Link Firewall on a network (lan-net) with public IP addresses. To make it possible to place a call from this phone to another H.323 phone on the Internet, and to allow H.323 phones on the Internet to call this phone, we need to configure firewall rules.

The following rules need to be added to the rule listings in the firewall, make sure there are no rules disallowing or allowing the same kind of ports/traffic before these rules.

WebUI :

1. Outgoing Rule

Rules → IP Rules → Add → IP Rule:

Enter the following:

Name: H323AllowOut

Action: Allow

Service: H323

Source Interface: LAN

Destination Interface: any

Source Network: lan-net

Destination Network: 0.0.0.0/0 (all-nets)

Comment: Allow outgoing calls.

Then click **OK**

2. Incoming Rule

Rules → IP Rules → Add → IP Rule:

Enter the following:

Name: H323AllowIn

Action: Allow

Service: H323

Source Interface: any

Destination Interface: LAN

Source Network: 0.0.0.0/0 (all-nets)

Destination Network: lan-net

Comment: Allow incoming calls.

Then click **OK**

Using Private IP Addresses

In this scenario a H.323 phone is connected to a D-Link Firewall on a network with private IP addresses. To make it possible to place a call from this phone to another H.323 phone on the Internet, and to allow H.323 phones on the Internet to call this phone, we need to configure firewall rules.

The following rules need to be added to the rule listings in the firewall, make sure there are no rules disallowing or allowing the same kind of ports/traffic before these rules. As we are using private IPs on the phone incoming traffic need to be SATed as in the example below. The object ip-phone below should be the internal IP of the H.323 phone.

WebUI :

1. Outgoing Rule

Rules → **IP Rules** → **Add** → **IP Rule**:

Enter the following:

Name: H323Out

Action: NAT

Service: H323

Source Interface: LAN

Destination Interface: any

Source Network: lan-net

Destination Network: 0.0.0.0/0 (all-nets)

Comment: Allow outgoing calls.

Then click **OK**

2. Incoming Rules

Rules → IP Rules → Add → IP Rule:

Enter the following:

Name: H323In

Action: SAT

Service: H323

Source Interface: any

Destination Interface: core

Source Network: 0.0.0.0/0 (all-nets)

Destination Network: ip-wan (external IP of the firewall)

Comment: Allow incoming calls to H.323 phone at ip-phone.

SAT

Translate Destination IP Address: To New IP Address: ip-phone (IP address of phone)

Then click **OK**

Rules → IP Rules → Add → IP Rule:

Enter the following:

Name: H323In

Action: Allow

Service: H323

Source Interface: any

Destination Interface: core

Source Network: 0.0.0.0/0 (all-nets)

Destination Network: ip-wan (external IP of the firewall)

Comment: Allow incoming calls to H.323 phone at ip-phone.

Then click **OK**

To place a call to the phone behind the D-Link Firewall, place a call to the external IP address on the firewall. If multiple H.323 phones are placed behind the firewall, one SAT rule has to be configured for each phone. This means that multiple external addresses have to be used. However, it is preferred to use a H.323 gatekeeper as in the "H.323 with Gatekeeper" scenario (see [18.4.5](#)), as this only requires one external address.



Example: Two Phones Behind Different D-Link Firewalls

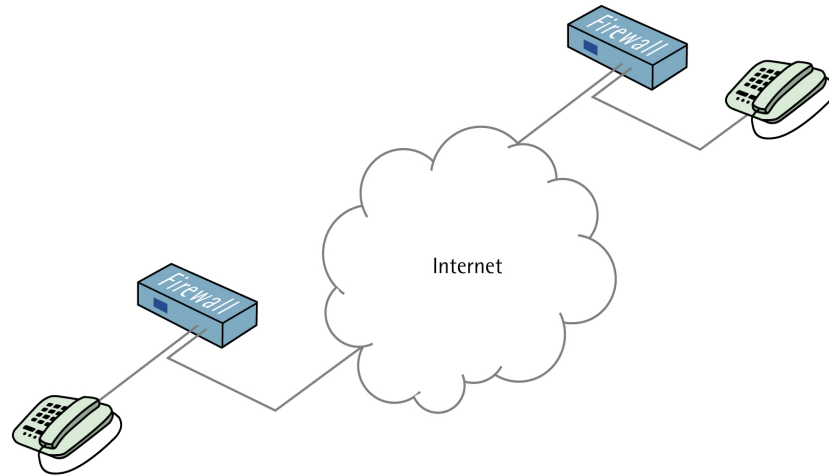


Figure 18.4: H.323 Scenario 2.

Using Public IP Addresses

This scenario consists of two H.323 phones, each one connected behind a D-Link Firewall on a network with public IP addresses. In order to place calls on these phones over the Internet, the following rules need to be added to the rule listings in both firewalls. Make sure there are no rules disallowing or allowing the same kind of ports/traffic before these rules.

WebUI :

1. Outgoing Rule

Rules → IP Rules → Add → IP Rule:

Enter the following:

Name: H323AllowOut

Action: Allow

Service: H323

Source Interface: LAN

Destination Interface: any

Source Network: lan-net

Destination Network: 0.0.0.0/0 (all-nets)

Comment: Allow outgoing calls.

Then click **OK**

2. Incoming Rule

Rules → IP Rules → Add → IP Rule:

Enter the following:

Name: H323AllowIn

Action: Allow

Service: H323

Source Interface: any

Destination Interface: LAN

Source Network: 0.0.0.0/0 (all-nets)

Destination Network: lan-net

Comment: Allow incoming calls.

Then click **OK**

Using Private IP Addresses

This scenario consists of two H.323 phones, each one connected behind a D-Link Firewall on a network with private IP addresses. In order to place calls on these phones over the Internet, the following rules need to be added to the rule listings in the firewall, make sure there are no rules disallowing or allowing the same kind of ports/traffic before these rules. As we are

using private IPs on the phones, incoming traffic need to be SATed as in the example below. The object ip-phone below should be the internal IP of the H.323 phone behind each firewall.

WebUI :

1. Outgoing Rule

Rules → IP Rules → Add → IP Rule:

Enter the following:

Name: H323Out

Action: NAT

Service: H323

Source Interface: LAN

Destination Interface: any

Source Network: lan-net

Destination Network: 0.0.0.0/0 (all-nets)

Comment: Allow outgoing calls.

Then click **OK**

2. Incoming Rules

Rules → IP Rules → Add → IP Rule:

Enter the following:

Name: H323In

Action: SAT

Service: H323

Source Interface: any

Destination Interface: core

Source Network: 0.0.0.0/0 (all-nets)

Destination Network: ip-wan (external IP of the firewall)

Comment: Allow incoming calls to H.323 phone at ip-phone.

SAT

Translate Destination IP Address: To New IP Address: ip-phone (IP address of phone)

Then click **OK**

Rules → IP Rules → Add → IP Rule:

Enter the following:

Name: H323In**Action:** Allow**Service:** H323**Source Interface:** any**Destination Interface:** core**Source Network:** 0.0.0.0/0 (all-nets)**Destination Network:** ip-wan (external IP of the firewall)**Comment:** Allow incoming calls to H.323 phone at ip-phone.Then click **OK**

To place a call to the phone behind the D-Link Firewall, place a call to the external IP address on the firewall. If multiple H.323 phones are placed behind the firewall, one SAT rule has to be configured for each phone. This means that multiple external addresses have to be used. However, it is preferred to use a H.323 gatekeeper as in the "H.323 with Gatekeeper" scenario (see 18.4.5), as this only requires one external address.


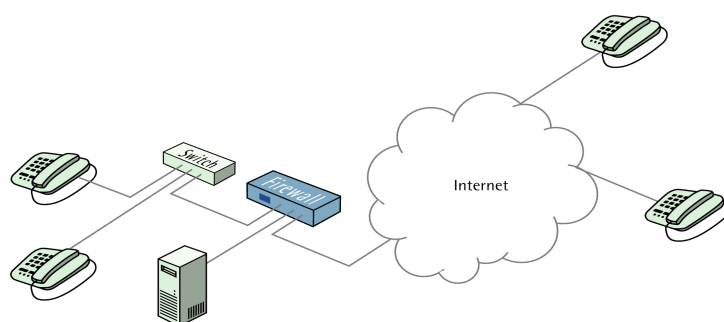
 **Example:** H.323 with Gatekeeper

Figure 18.5: H.323 Scenario 3.

In this scenario, a H.323 gatekeeper is placed in the DMZ of the D-Link Firewall. A rule is configured in the firewall to allow traffic between the

private network where the H.323 phones are connected on the internal network and to the Gatekeeper on the DMZ. The Gatekeeper on the DMZ is configured with a private address.

The following rules need to be added to the rule listings in both firewalls, make sure there are no rules disallowing or allowing the same kind of ports/traffic before these rules.

WebUI :

1. Incoming Gatekeeper Rules

Rules → IP Rules → Add → IP Rule:

Enter the following:

Name: H323In

Action: SAT

Service: H323-Gatekeeper

Source Interface: any

Destination Interface: core

Source Network: 0.0.0.0/0 (all-nets)

Destination Network: ip-wan (external IP of the firewall)

Comment: SAT rule for incoming communication with the Gatekeeper located at ip-gatekeeper.

SAT

Translate Destination IP Address: To New IP Address: ip-gatekeeper (IP address of gatekeeper)

Then click **OK**

Rules → IP Rules → Add → IP Rule:

Enter the following:

Name: H323In

Action: Allow

Service: H323-Gatekeeper

Source Interface: any

Destination Interface: core

Source Network: 0.0.0.0/0 (all-nets)

Destination Network: ip-wan (external IP of the firewall)

Comment: Allow incoming communication with the Gatekeeper.

Then click **OK**

Rules → IP Rules → Add → IP Rule:

Enter the following:

Name: H323In

Action: Allow

Service: Gatekeeper

Source Interface: LAN

Destination Interface: DMZ

Source Network: lan-net


Destination Network: ip-gatekeeper (IP address of gatekeeper)

Comment: Allow incoming communication with the Gatekeeper.

Then click **OK**



- There is no need to specify a specific rule for outgoing calls. The D-Link Firewall monitors the communication between "external" phones and the Gatekeeper to make sure that it is possible for internal phones to call the external phones that are registered with the gatekeeper. ■

 **Example:** H.323 with Gatekeeper and two D-Link Firewalls

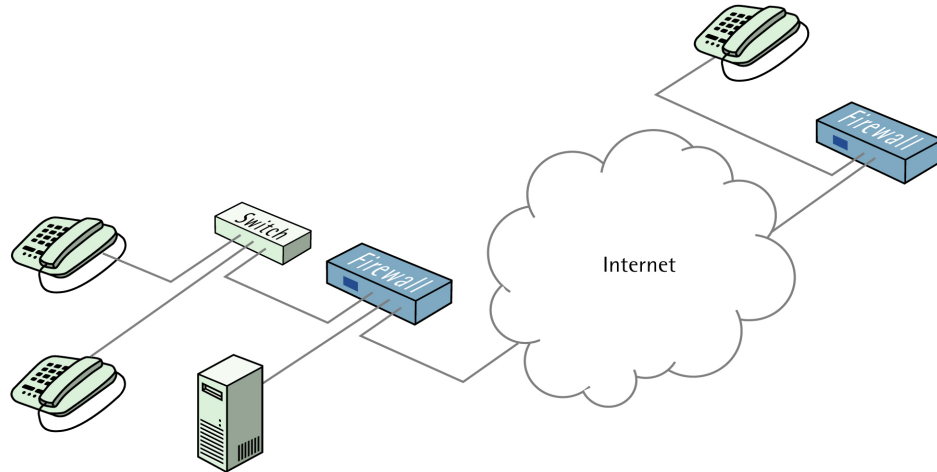


Figure 18.6: H.323 Scenario 4.

This scenario is quite similar to scenario 3, with the difference of a D-Link Firewall protecting the "external" phones. The D-Link Firewall with the Gatekeeper connected to the DMZ should be configured exactly like in scenario 3 (see [18.4.5](#)). The other D-Link Firewall should be configured as follows.

The following rules need to be added to the rule listings in the firewall, make sure there are no rules disallowing or allowing the same kind of ports/traffic before these rules.

WebUI :

1. Outgoing Gatekeeper Rule

Rules → **IP Rules** → **Add** → **IP Rule:**

Enter the following:

Name: H323Out

Action: NAT

Service: H323-Gatekeeper

Source Interface: LAN

Destination Interface: any

Source Network: lan-net

Destination Network: 0.0.0.0/0 (all-nets)

Comment: Allow outgoing communication with a gatekeeper.

Then click **OK**



- There is no need to specify a specific rule for outgoing calls. The D-Link Firewall monitors the communication between "external" phones and the Gatekeeper to make sure that it is possible for internal phones to call the external phones that are registered with the gatekeeper. ■



Example: Using the H.323 ALG in a Corporate Environment

This scenario is an example of a more complex network that shows how the H.323 ALG can be deployed in a corporate environment. At the head office DMZ a H.323 Gatekeeper is placed that can handle all H.323 clients in the head-, branch- and remote offices. This will allow the whole corporation to use the network for both voice communication and application sharing. It is assumed that the VPN tunnels are correctly configured and that all offices use private IP-ranges on their local networks. All outside calls are done

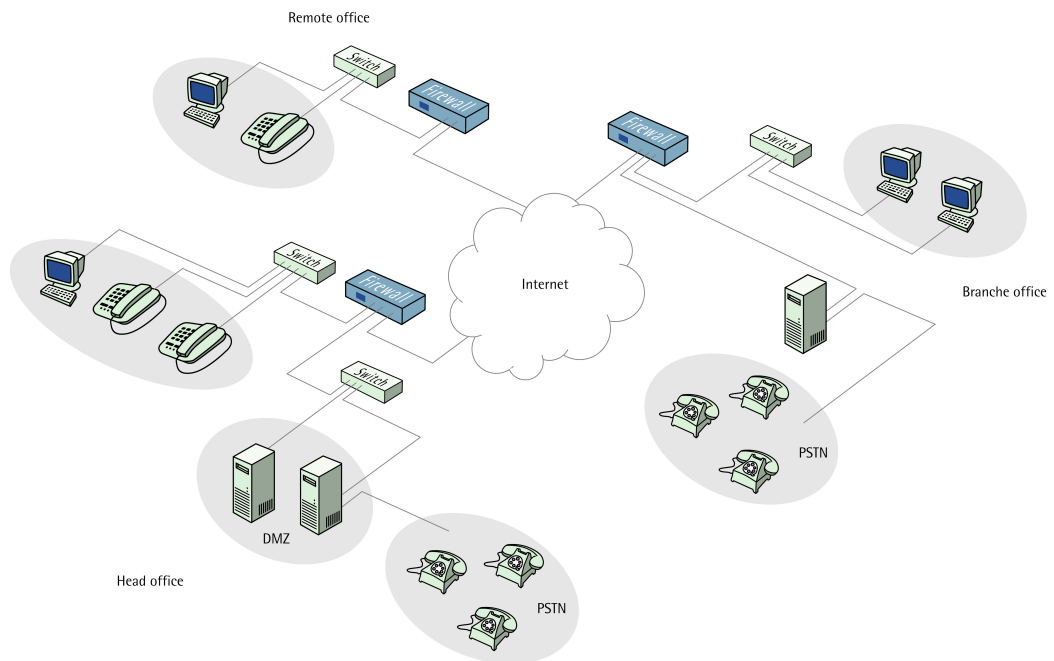


Figure 18.7: H.323 Scenario 5.

over the existing telephone network using the gateway (ip-gateway) connected to the ordinary telephone network.

Head Office Firewall Configuration

The head office has placed a H.323 Gatekeeper in the DMZ of the corporate D-Link Firewall. This D-Link Firewall should be configured as follows.

WebUI :

1. **Rules → IP Rules → Add → IP Rule:**

Enter the following:

Name: LanToGK

Action: Allow

Service: Gatekeeper

Source Interface: LAN

Destination Interface: DMZ

Source Network: lan-net

Destination Network: ip-gatekeeper

Comment: Allow H.323 entities on lan-net to connect to the Gatekeeper.

Then click **OK**

2. **Rules → IP Rules → Add → IP Rule:**

Enter the following:

Name: LanToGK

Action: Allow

Service: H323

Source Interface: LAN

Destination Interface: DMZ

Source Network: lan-net

Destination Network: ip-gateway

Comment: Allow H.323 entities on lan-net to call phones connected to the H.323 Gateway on the DMZ. Remember to use the correct service.

Then click **OK**

3. Rules → IP Rules → Add → IP Rule:

Enter the following:

Name: GWToLan

Action: Allow

Service: H323

Source Interface: DMZ

Destination Interface: LAN

Source Network: ip-gateway

Destination Network: lan-net

Comment: Allow communication from the Gateway to H.323 phones on int-net. Remember to use the correct service.

Then click **OK**

4. Rules → IP Rules → Add → IP Rule:

Enter the following:

Name: BranchToGW

Action: Allow

Service: H323-Gatekeeper

Source Interface: vpn-branch

Destination Interface: DMZ

Source Network: branch-net

Destination Network: ip-gatekeeper, ip-gateway

Comment: Allow communication with the Gatekeeper on DMZ from the Branch network

Then click **OK**

5. Rules → IP Rules → Add → IP Rule:

Enter the following:

Name: BranchToGW

Action: Allow

Service: H323-Gatekeeper

Source Interface: vpn-remote

Destination Interface: DMZ

Source Network: remote-net

Destination Network: ip-gatekeeper

Comment: Allow communication with the Gatekeeper on DMZ from the Remote network

Then click **OK**

Branch and Remote Office Firewall

The branch and remote office H.323 phones and applications will be configured to use the H.323 Gatekeeper at the head office. The D-Link Firewalls in the remote and branch offices should be configured as follows.

The following rule should be in both the Branch and Remote Office firewalls.

WebUI :

1. **Rules** → **IP Rules** → **Add** → **IP Rule**:

Enter the following:

Name: ToGK

Action: Allow

Service: H323-Gatekeeper

Source Interface: LAN

Destination Interface: vpn-hq

Source Network: lan-net

Destination Network: hq-net

Comment: Allow communication with the Gatekeeper connected to the Head Office DMZ.

Then click **OK**

The branch office D-Link Firewall has a H.323 Gateway connected to its DMZ. In order to allow the Gateway to register with the H.323 Gatekeeper at the Head Office, the following rule has to be configured.

WebUI :

1. **Rules** → **IP Rules** → **Add** → **IP Rule**:

Enter the following:

Name: GWToGK

Action: Allow

Service: H323-Gatekeeper

Source Interface: DMZ

Destination Interface: vpn-hq

Source Network: ip-branchgw

Destination Network: hq-net

Comment: Allow the Gateway to communicate with the Gatekeeper connected to the Head Office.

Then click **OK**

■  **Note** ■

- There is no need to specify a specific rule for outgoing calls. The D-Link Firewall monitors the communication between "external" phones and the Gatekeeper to make sure that it is possible for internal phones to call the external phones that are registered with the gatekeeper. ■

CHAPTER 19

Intrusion Detection System (IDS)

19.1 Overview

Intrusion Detection is a technology that monitors network traffic, searching for signs of security violations, or *intrusions*. An intrusion can be defined as an attempt to compromise certain parts of a computer system, or to bypass its security mechanisms. As these forms of attacks are a common occurrence on the Internet, and can often be easily automatized by an attacker, Intrusion Detection is an important technology to identify and prevent these threats.

In order to make an effective and reliable IDS, D-Link IDS goes through three levels of processing and addresses the following questions:

- What traffic to analyze
- What to search for (i.e. what is an "attack")
- What action to carry out

As an example, picture a system that is monitoring FTP. It would only be concerned with traffic relating to FTP, while traffic relating to, for example POP3, would be of no interest what so ever. Also, only attacks that refer to the FTP protocol would be of interest.

D-Link IDS uses a combination of *Intrusion Detection Rules*, *Pattern Matching*, and *Actions*, in order to answer the three questions mentioned above.

19.1.1 Intrusion Detection Rules

An Intrusion Detection Rule defines the kind of traffic – service – that should be analyzed. Filtering fields regarding source and destination interfaces, networks, ports, and protocols are also defined here. Only traffic matching this rule is passed on to the next processing level of IDS, where actual analysis takes place.

19.1.2 Pattern Matching

In order for the IDS to correctly identify an attack, it has to know what an attack is. To achieve this, pre-defined patterns, called "signatures", are created that describe certain attacks. The network traffic is then analyzed by the IDS, searching for these patterns. This is also known as "misuse detection" or "signature detection".

Consider the following example. A user tries to retrieve the password file "passwd" from a system, using FTP:

```
RETR passwd
```

A signature looking for the ASCII text strings "RETR" and "passwd" would cause a match in this case, signalling that an attack has been found.

In order to make this example easy to follow, patterns containing ASCII text strings was used. This is not necessary; patterns can just as well contain binary data.

If an attack is found, the next processing level of the IDS is carried out – cause of action.

19.1.3 Action

After an intrusion has been detected, an action, or response, must be taken. Depending on the severity of the attack, traffic can either be dropped, logged, both, or simply ignored.

19.2 Chain of Events

The following is a simplified picture of the chain of events when a packet arrives on the firewall, with focus on the IDS (note that no other sub-systems are considered here). Two scenarios are possible, one where the firewall rule set has to accept the packet before passing it on to the IDS, and one where the IDS can process traffic even if the rule set decides that the packet should be dropped.

19.2.1 Scenario 1

Traffic is only passed on to the IDS if the firewall's IP rule set decides that it is valid, shown in Figure 19.1.

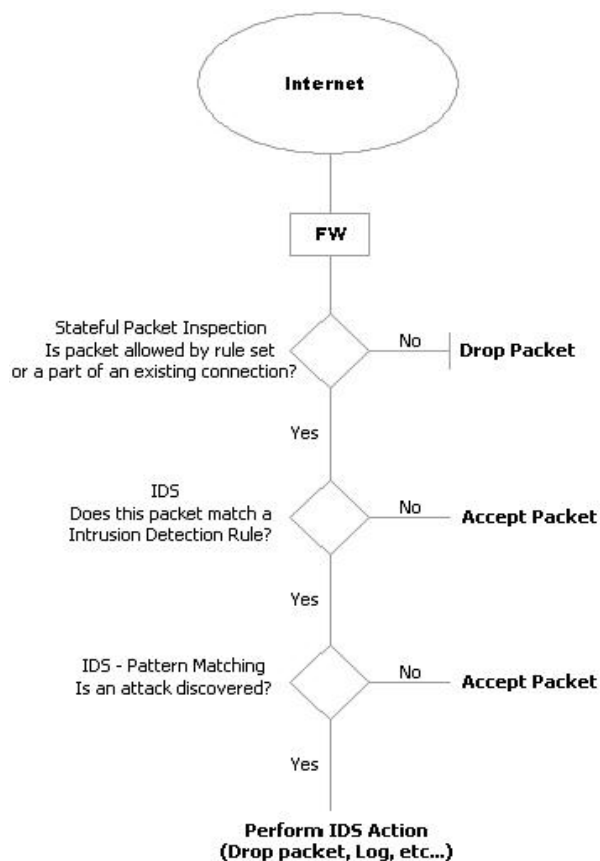


Figure 19.1: IDS Chain of Events Scenario 1

1. A packet arrives on the firewall and initial verifications regarding source/destination IP addresses and source/destination ports are performed. If this packet is accepted by the firewall's IP rule set, a connection will be established between the source and destination, before passing the packet on to the IDS sub-system. If the packet is a part of an already existing connection, it is also passed on to the IDS sub-system. If the packet is denied by the IP rule set, it is dropped.
2. The source and destination information of the packet is compared to the Intrusion Detection Rules. If a match is found, it is passed on to the next level of IDS processing - pattern matching. If not, it is accepted, with possible further actions, as defined by the rule set (for example address translation, logging, etc).
3. The pattern-matching engine searches the payload of the packet for pre-defined signatures. If a match is found, the final level of IDS processing will be carried out – the action. If not, the packet is accepted, with possible further actions, as defined by the rule set (for example address translation, logging, etc).
4. Depending on the action defined in the Intrusion Detection Rule, the packet can be dropped, logged, both, or ignored.

19.2.2 Scenario 2

This is similar to the first scenario, but with one big difference. Traffic will always be passed on to the IDS regardless of the action chosen by the firewall's IP rule set. This means that traffic that the firewall drops will also be analyzed. Figure 19.2 shows the events sequence when the firewall's IP rule set decides that the traffic is not valid and should be dropped and the traffic is passed to IDS for further analyzing.

1. A packet arrives on the firewall and initial verifications regarding source/destination IP addresses and source/destination ports are performed. The firewall's IP rule set decides that this packet should be dropped, but before that, traffic is passed on to the IDS sub-system for further analyzing.
2. The source and destination information of new packet is compared to the Intrusion Detection Rules. If a match is found, it is passed on to the next level of IDS processing - pattern matching. If not, the packet is dropped.

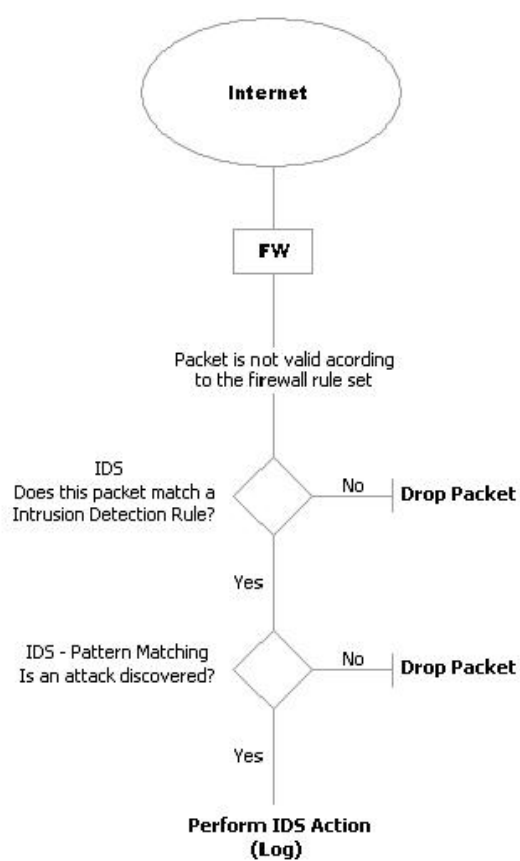


Figure 19.2: IDS Chain of Events Scenario 2

3. The pattern-matching engine searches the payload of the packet for pre-defined signatures. If a match is found, the final level of IDS processing is carried out – the action. If not, the packet is dropped.
4. As this packet will not be accepted by the firewall, the only interesting action is to log the attempted intrusion.

19.3 Signature Groups

Usually, several attacks exist for a specific protocol, and it would be most favorable to search for all of them at the same time when analyzing network traffic. To do this, signatures that refer to the same protocol are grouped together. For example, all signatures that refer to the FTP protocol are located in one group, while signatures that refer to POP3 are located in another group. In addition to this, signatures that originate from the same source are also grouped together. This means that signatures that are only valid when originating from the external network are grouped together, while signatures that are valid when originating from the internal network are located in another group. This is done in order to allow more effective processing for the IDS.

19.4 Automatic Update of Signature Database

Discovering new attacks is an ongoing process. New attacks are sometimes discovered daily, so it is important to have an up-to-date signature database in order to protect the network from the latest threats. The signature database contains all signatures and signature groups currently recognized by the IDS.

A new, updated signature database can be automatically downloaded by the firewall, at a configurable interval. This is done through a HTTP connection to a D-Link server, hosting the latest signature database file. If this signature database file has a newer version than the current, the new signature database will be downloaded, thus replacing the old version. This will ensure that the signature database is always up-to-date.

Figure 19.3 is a simplified picture that describes the communication flow when a new signature database file is downloaded:

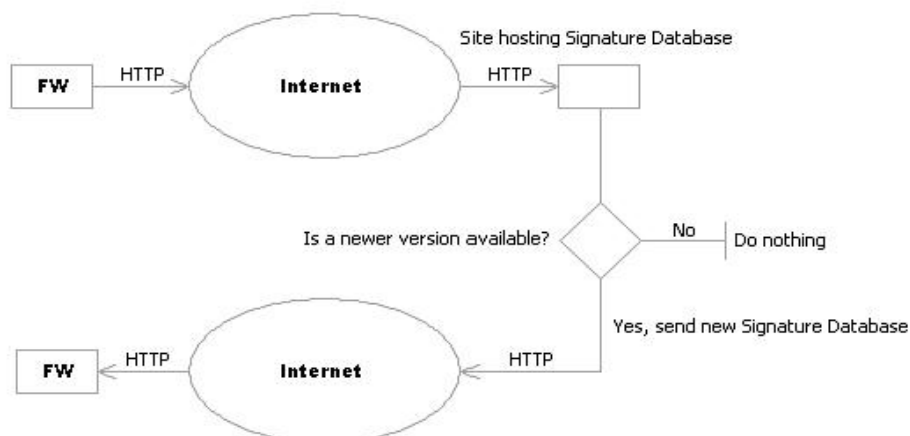



Figure 19.3: Signature Database Update

19.5 SMTP Log Receiver for IDS Events

In order to receive notifications via e-mail of IDS events, a SMTP Log receiver can be configured. This e-mail will contain a summary of IDS events that has occurred in a user-configurable period of time.

When an IDS event has occurred, the D-Link firewall will wait for **Hold Time** seconds before sending the notification e-mail. However, the e-mail will only be sent if the number of events occurred in this period of time is equal to, or bigger, than **Log Threshold**. When this e-mail has been sent, the firewall will wait for **Minimum Repeat Time** seconds before sending a new e-mail.

 **Example:** Configuring a SMTP Log Receiver

In this example, an Intrusion Detection Rule is configured with a SMTP Log Receiver and the following values:

Minimum Repeat Time: 600 seconds
 Hold Time: 120 seconds
 Log Threshold: 2 events

Once an IDS event occurs, the Intrusion Detection Rule is triggered. At least one new event occurs within the Hold Time, 120 seconds, thus reaching the log threshold level (at least 2 events has occurred). This results in an e-mail to be sent, containing a summery of the IDS events. Several more IDS events may occur after this, but to prevent flooding the mail server, the firewall will wait for 600 seconds (10 minutes) before sending a new e-mail, containing information about the new events. A SMTP server is assumed to have been configured in the address book, with an IP address object name "smtp-server".

WebUI :

1. SMTP log receiver:

- adding a SMTP log receiver

System → **Log and Event Receivers** → **Add**

→ **SMTP Event Receiver:**

General

Enter the following:

Name: smtp4IDS

SMTP Server: smtp-server

Server Port: 25 (by Internet standard)

Fill in alternative e-mail addresses in the edit boxes(up to 3 addresses can be configured).

Sender: hostmaster

Subject: Log event from D-Link Firewall

Minimum Repeat Delay: 600

Hold Time: 120

Log Threshold: 2

Then click **OK**.

2. IDS Rules:

- Enabling logging in the "**Log Settings**" configuration page for a specific IDS rule and using **All receivers** or specific receiver "**smtp4IDS**" configured above as log receiver.

19.6 Scenario: Setting up IDS

The following example illustrates the steps needed to set up IDS for a simple scenario where a mail server is exposed to the Internet on the DMZ network, with a public IP address, and is to be protected by the IDS, as shown in Figure 19.4. The Internet can be reached through the firewall on the WAN interface.

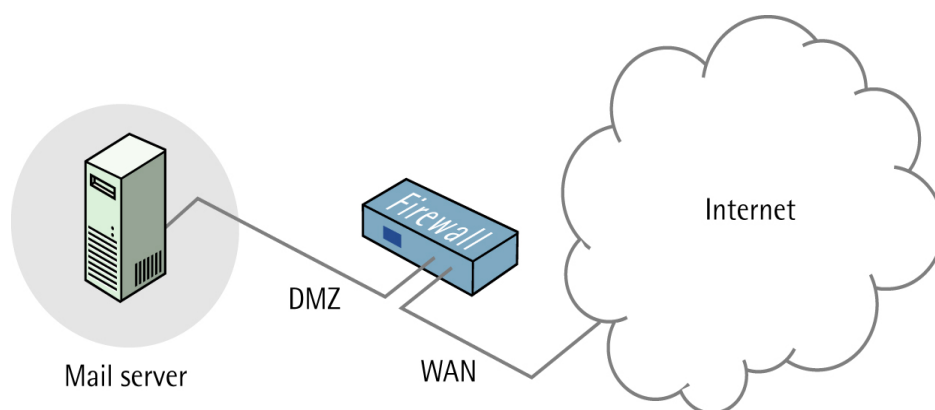


Figure 19.4: An IDS Scenario

WebUI :

1. Configuring Objects and Services:

It is assumed that an object defining the mail server has been created, and that interface and network objects exist for the internal and external network.

In case a service for SMTP does not already exist, it must be created, which is done in **Objects** → **Services**. Type is TCP, and destination port is 25.

2. Create IDS Rule

This IDS rule will be called IDSMailSrvRule, and the service to use is the previously created SMTP service. Source Interface and Source Network defines where traffic is coming from, in this example the external network. The Destination Interface and Destination Network define where traffic is directed to, in this case the mail server. Destination Network should therefore be set to the object defining the mail server.

IDS/IDP → IDS Rules → Add → IDS/IDP Rule:**Name:** IDSMailSrvRule**Service:** smtp

Also inspect dropped packets: In case all traffic matching this rule should be scanned (this also means traffic that the main rule-set would drop), the "Also inspect dropped packets" checkbox should be checked, which is the case in this example.

Source Interface: WAN**Source Network:** wan-net**Destination Interface:** DMZ**Destination Network:** ip_mailserverThen click **OK**

If logging of intrusion attempts is desired, this can be configured in the Logging tab, where log receivers can be chosen.

3. Create IDS Action

When this IDS Rule has been created, an action must also be created, specifying what signatures the IDS should use when scanning data matching the IDS Rule, and what the firewall should do in case an intrusion is discovered. Intrusion attempts should cause the connection to be dropped, so "Action" is set to Protect. Severity is set to All, in order to match all possible signatures. "Signatures" is set to FROM_EXT_MAIL_SMTP, in order to use signatures that describe attacks from the external network, dealing with the SMTP protocol.

IDS/IDP → IDS Rules → IDSMailSrvRule → Add → IDS Rule Action:**Action:** Protect**Severity:** All**Signatures:** FROM_EXT_MAIL_SMTPThen click **OK**

To summarize, the following will occur: If traffic from the external network, to the mail server is discovered, the IDS will be activated. In case the traffic matches any of the signatures in the FROM_EXT_MAIL_SMTP signature group, the connection will be dropped, thus protecting the mail server. If a log receiver has been configured, the intrusion attempt will also be logged.

Part VIII

**Virtual Private Network
(VPN)**

VPNs, *Virtual Private Networks*, provide means of establishing secure links to parties. It is extended over public networks via the application of *Encryption* and *Authentication*, offering good flexibility, effective protection, and cost efficiency on connections over the Internet.

Topics in this part includes:

- [Introduction to VPN](#)
- [Introduction to Cryptography](#)
- [VPN in Firewalls](#)
- [VPN Protocols & Tunnels](#)
- [VPN Planning](#)

CHAPTER 20

VPN Basics

20.1 Introduction to VPN

Long gone is the time when corporate networks were separate isles of local connectivity. Today, most networks are connected to each other by the Internet. Issues of protecting the local networks from Internet-based crime and intrusion are being solved by firewalls, intrusion detection systems, anti-virus software and other security investments. However, business is increasingly often being done across the Internet as a means of efficient and inexpensive communication.

As we all have learned the hard way, not all parts of the Internet can be trusted in our time. Private interests as well as corporate communication requirements necessitate a means for data to be able to travel across the Internet to its intended recipient without allowing anyone else to read or alter it. It is equally important that the recipient can verify that no one is falsifying information, i.e. pretending to be someone else.

VPNs, Virtual Private Networks, provide a very cost efficient means of establishing secure links to parties that wish to exchange information in a trustworthy manner.

20.1.1 VPNs vs Fixed Connections

Using leased lines or other non-public channels to exchange data between organizations is not a new concept. It has been done since the first

computers began talking to each other. In the beginning, communication was limited to local area communication links, but in time, people were finding reasons to have their computers exchange information across greater distances.

Fixed connections are usually very reliable as far as uptime and available bandwidth is concerned. They are also fairly secure, as long as no one attacks the telephony infrastructure or digs your optical fibres out of the ground and attach their own equipment to it. Fixed long-distance connections, provided that suitable security measures are taken, may be considered "Private Networks".

However, fixed channels of communication are just that: fixed. If you hire a fixed connection between company A and B, you only allow communication between companies A and B. If several organizations would want to communicate with each other in all directions, separate fixed connections between all organizations would be needed. Such situations quickly escalate beyond all manageability and cost efficiency:

- *Two organizations only require 1 connection.*
- *Three organizations require 3 connections.*
- *Five organizations require 10 connections.*
- *Seven organizations require 21 connections.*
- *Ten organizations require 45 connections.*
- *100 organizations require 4950 connections.*

One could argue that maybe some communication could be done by the way of intermediates. If I wish to talk to company B, maybe I can send my data to company C that has a link to company B? That way I don't have to have a link to company B of my own?

In some cases, and in a small scale, this may work. On the other hand, it may not work at all even if it is on a manageable scale. Consider a company that sells a product to ten customers who all compete with each other.

- Would any one of them accept that their orders and delivery confirmations travel through the hands of one of their competitors?
- Hardly.

Another solution is required.

From a connectivity and security perspective, Virtual Private Networks may still be viewed as "fixed connections" in that they do provide connectivity between two or more organizations. This is a fact that does not change even though *Cryptography* is deployed to implement the "Virtual" side of the "Private Network".

20.2 Introduction to Cryptography

Cryptography provides a means to create "Virtual Private Networks" across the Internet with no additional investments in cables, leased lines, or other connectivity. It is an umbrella expression covering three basic techniques and benefits:

Confidentiality

No one but the intended recipients is able to intercept and understand the communication. Confidentiality is accomplished by encryption.

Authentication & Integrity

Proof for the recipient that the communication was actually sent by the expected sender, and that the data has not been modified in transit. This is accomplished by authentication, often by use of cryptographic keyed hashes.

Non-repudiation

Proof that the sender actually sent the data; the sender cannot later deny having sent it. Non-repudiation is usually a benign side-effect of authentication.

20.2.1 Encryption

Encryption is a process of encoding sensitive information from *plaintext* to unreadable *ciphertext* through some mathematical algorithms. The operation of the algorithms is varied and usually parameterized by a large random number, known as a *key*. The ciphertext is encrypted by the key and it needs the same key or a related key to perform the reverse procedure – *decryption*, to return to the original plaintext.

The algorithms of Encryption can be categorized into three types – *symmetric*, *asymmetric*, and *hybrid* encryption.

Symmetric Encryption

In symmetric encryption, the same key is used for both encryption and decryption. Therefore the key is shared by the sender and the recipients, and must be kept secretly. Using the same secret key is a faster and simpler computation method, but the key distribution among users in the first place is a major problem, which must be carried out very carefully to prevent from passing the key to a wrong hand.

To secure the sharing of the secret key, *session keys* or *public keys* are often involved in the actual operation.

A session key, as its name describes, is only valid for one session. Even if the key is compromised at a session, it cannot be used for future decryption. Another solution is the use of public key handled by *asymmetric encryption* presented next.

Currently, common used symmetric encryption algorithms include:

- DES and Triple DES
 - DES uses a 56-bit key and is considered equal in strength to most other algorithms that use 40-bit keys. Its relatively short key length by modern standards means that it is now considered vulnerable to *brute force attacks*.

Triple-pass DES uses three different keys in three DES passes, forming a theoretical key length of 168 bits.

- Blowfish
 - A 64-bit block cipher with key length variable between 32 and 448 bits.
- Twofish
 - A 128-bit block cipher with key length 128, 192, or 256 bits.
- CAST-128
 - A 64-bit block cipher with a 128-bit key, less frequently employed than Blowfish.
- AES
 - A 128-bit block size with key lengths of 128-256 bits, a sound alternative to the ageing DES.

D-Link firewall's VPN implementation supports all the above algorithms.

Asymmetric Encryption

A pair of keys is used in asymmetric encryption, one called a *public key*, which can be available to anyone who wants to use encryption, and the other, called a *private key*, that must be kept confidentially and is known only by the owner.

The two keys are very large prime numbers and mathematically related, but one can not be used for resolving the other. Anyone can send private information to a recipient, say *A*, by encrypting the information using *A*'s *public key*. But only *A* will be able to recover the information by decrypting the ciphertext using the related *private key*. Moreover, if some known information can be correctly recovered by decrypting with *A*'s public key, it must have been encrypted with *A*'s private key, and therefore by *A*. This means that asymmetric algorithms provide proof of origin. *RSA* and *DSA* are the most well-known and most commonly-used asymmetric algorithms.

Compared to symmetric encryption, the much longer keys cause slower speed and intensive resource use to asymmetric encryption, and hence unsuitable for encrypting large quantity of data. It is generally used for aiding the symmetric key distribution and authentication tasks. The combination of symmetric and asymmetric algorithms is called Hybrid Encryption.

Hybrid Encryption

The hybrid encryption combines the best of the two worlds: symmetric and asymmetric algorithms. The symmetric key provides the fastest encryption and decryption, and the asymmetric scheme provides a convenient way to share the secret key.

Diffie-Hellman key exchange protocol:

The Diffie-Hellman protocol allows users to exchange a secret key over an insecure medium without any prior secrets, which is one of the most widely used key exchange methods supporting various secure Internet protocols, e.g. SSL, SSH, and IPsec.

In the protocol, each side of the connection generates a related private-public key pair, and publishes the public part. After the public key exchange, one is able to compute a new secret key using one's private key and the other's public key. The resulting key is common to both sides, and can be used as a shared secret key for symmetric encryption. In such a way,

the critical keying information is not transmitted through the insecure connection.

20.2.2 Authentication & Integrity

In addition to encryption, *Authentication* methods are necessary to ensure the integrity and authenticity of encrypted data.

One might easily think that encryption provides good enough protection; it does after all ensure that the information is transferred in unreadable ciphertext. However, encryption does not provide any sort of protection against alteration of the encrypted data and nothing about the user's identity.

If someone can intercept the encrypted data stream and modify it, the result on the receiving end, after decryption, would also be altered. The end result of the modifications would certainly be unpredictable to the person intercepting the data stream, but if his goal is to harm in subtle ways, modification of the encrypted data may certainly be enough. What if, for instance, a document is sent for printing in hundreds of thousands of copies, and the text is garbled on every tenth page?

Another undesired case is the so called *man-in-the-middle attack*, where a third party intercepts the public keys from the exchange of 2 sides and reply by bogus keys. This way, the man in the middle establishes 2 secured connections to both sides, and can decrypt their conversations freely.

These cases are where authentication mechanism comes into play. Authentication serves to prove to the recipient that the data was actually sent by the person claiming to have sent it. And more importantly, it proves that the data has not been altered after leaving the sender. The mechanism is accomplished by the use of *Digital Signature* and *Certificate*.

Digital Signature

A digital signature is a stamp that is used to prove the identity of one person, and to ensure the integrity of the original message. The signature is created using the *asymmetric encryption* scheme; it cannot be imitated by someone else, and the sender cannot easily repudiate the message that has been signed.

The procedure of producing a digital signature works as follows:

On the sender's side:

- The sender prepares a *public-private key pair*, and publishes the public one.
- A one way function, known as *hash function*, is operated on a message, and a fixed length *message digest* is obtained. (The mathematical function is only one-way; the original message cannot be recomputed from the digest and any change to the original message will make the digest totally different.)
- The sender encrypts the message digest using the *private key*.
- The encrypted message digest becomes the sender's digital signature of the message, and is *unique* to that message.
- The digital signature is sent to the receiver together with the original plaintext message.

On the other side:

- The receiver uses the *hash function* to make a message digest of the received plaintext message.
- Using the *sender's public key*, the receiver decrypts the *digital signature* to get the sender computed message digest.
- The two digests are compared.
- If the two digests are identical, the received message is valid.

Certificate

As it is introduced in [8.4 X.509 Certificates](#), D-Link firewalls also support the *digital certificate* to be used to further authenticate that the public key really belongs to the alleged party.

A certificate is issued by a certification authority (CA) containing a copy of the certificate holder's *public key* and corresponding information, a serial number, expiration time, and the *digital signature* of the CA, so that a recipient can verify that the certificate is real. The digital certificates supported by D-Link firewalls conform to X.509 standard.

The CA creates the certificate by signing the authenticating *public key* and the *identity information* of the key holder with its own *private key*. The recipients have copies of *CA's public key* to be able to validate the certificate signature and trust the CA and the signed public key.

The CAs are also responsible for managing the CRLs to report the certificates that are no longer valid because, for example, the corresponding private key is compromised or the identity information has changed.

20.3 Why VPN in Firewalls

Virtual Private Network (VPN) may be implemented in many different ways. The greatest differences lie in whether or not to use security gateways: network devices whose purpose is to perform the work of *encryption* and *authentication*. There are both benefits and drawbacks of every different security gateway deployment.

The security gateway, may be placed in several different locations in relation to your *border router* and your *firewall*:

- Outside the firewall, in-line
- Outside the firewall, on the external network
- Between the firewall and the internal network
- On the internal network
- In a separate DMZ
- Incorporated in the firewall itself ✓

Each scenario above has its distinct benefits and drawbacks. Issues that need to be considered include:

- Can the firewall protect the security gateway and log attempted attacks on it?
- Does the configuration support roaming clients?
- Can the firewall inspect and log traffic passing in and out of the VPN?
- Does the configuration add points of failure to the Internet connection?

- In cases where the VPN gateway is located outside the firewall, can the firewall recognize VPN protected traffic from plaintext Internet traffic, so that it knows what to pass through to the internal network?
- Does it require additional configuration to the firewall or hosts participating in the VPN?

In D-Link firewalls, the *Security Gateway VPN* is integrated in the firewall itself. The reasons for this design can be found in the scenario analysis presented next.

20.3.1 VPN Deployment

Outside the Firewall, In-line

(Figure 20.1)

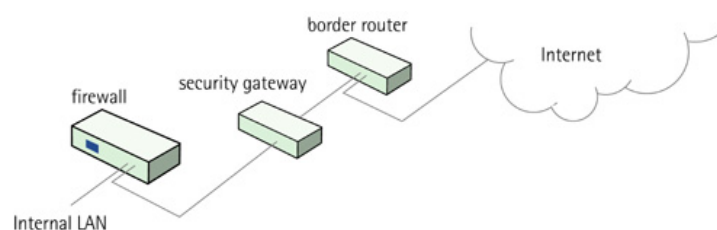


Figure 20.1: VPN Deployment Scenario 1

◇ *Benefits*

- Supports roaming clients, although it is difficult
- No special routing information is needed in the firewall
- The firewall can inspect and log plaintext from the VPN

◇ *Drawbacks*

- The Security Gateway is not protected by the firewall
- The firewall cannot easily determine which traffic came through an authenticated VPN and which came from the Internet, especially in the case of roaming clients
- Internet connectivity depends on the Security Gateway

Outside the Firewall, on the External Network

(Figure 20.2)

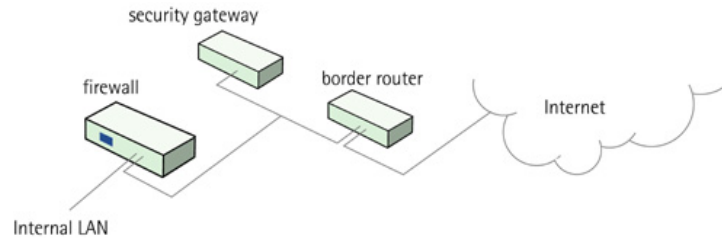


Figure 20.2: VPN Deployment Scenario 2

◇ *Benefits*

- Internet connectivity does not depend on Security Gateway
- The firewall can inspect and log plaintext from the VPN

◇ *Drawbacks*

- The Security Gateway is not protected by the firewall
- The firewall cannot easily determine which traffic came through an authenticated VPN and which came from the Internet, unless the border router can be trusted to do extensive filtering
- Special routing information is needed in the firewall
- Support for roaming clients is nearly impossible

Between the Firewall and the Internal Network

(Figure 20.3)

◇ *Benefits*

- Supports roaming clients
- No special routing information needed in the firewall
- The firewall can protect the Security Gateway

◇ *Drawbacks*

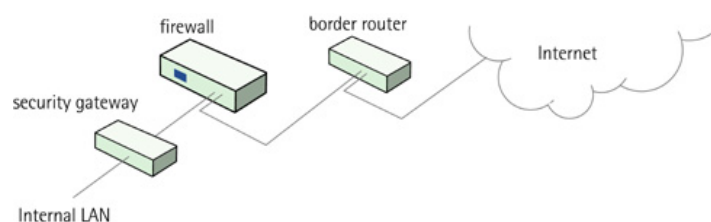


Figure 20.3: VPN Deployment Scenario 3

- Internet connectivity depends on the Security Gateway
- The firewall cannot inspect nor log plaintext from the VPN

VPN traffic should not normally be considered to be an integrated part of the internal network.

On the Internal Network

(Figure 20.4)

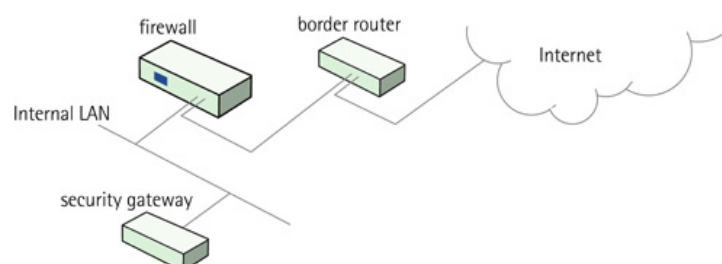


Figure 20.4: VPN Deployment Scenario 4

◇ *Benefits*

- The firewall can protect the Security Gateway
- Internet connectivity does not depend on the Security Gateway

◇ *Drawbacks*

- The firewall cannot inspect nor log plaintext from the VPN
- Special routes need to be added to the firewall, or to all internal clients participating in the VPN
- Support for roaming clients is very hard to achieve

In a separate DMZ

(Figure 20.5)

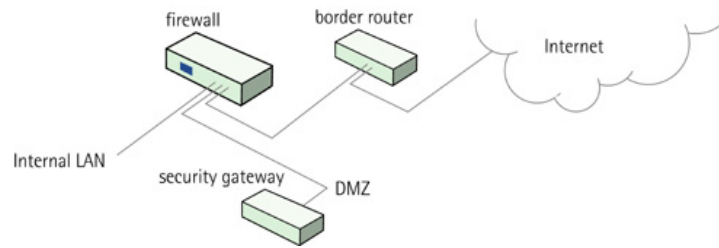


Figure 20.5: VPN Deployment Scenario 5

◇ *Benefits*

- The firewall can protect the Security Gateway
- Internet connectivity does not depend on the Security Gateway
- The firewall can inspect and log plaintext from the VPN

◇ *Drawbacks*

- Special routes need to be added to the firewall
- Support for roaming clients is very hard to achieve, since the firewall will not know to route through the Security Gateway in order to reach the VPN clients with moving IPs

Incorporated in the Firewall

(Figure 20.6)

◇ *Benefits*

- The firewall can protect the Security Gateway subsystem
- The firewall can inspect and log plaintext from the VPN
- Supports roaming clients

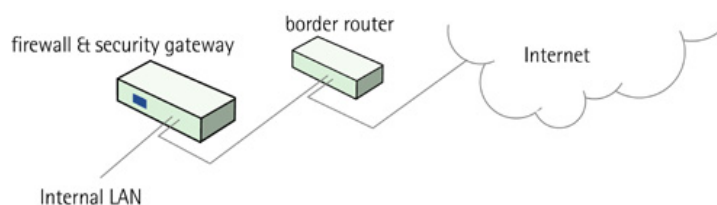


Figure 20.6: VPN Deployment Scenario 6

- No special routes need to be added to hosts participating in the VPN
- Can seamlessly integrate VPN and firewall policies

◇ *Drawbacks*

- The integrated Security Gateway may make the firewall less stable. However, it does not add another piece of hardware to the chain of points that may fail.

This solution provides the highest degree of functionality & security and is chosen by D-Link's design. All normal modes of operation are supported, and all traffic may be inspected and logged by the firewall.

CHAPTER 21

VPN Planning

21.1 VPN Design Considerations

”A chain is never stronger than its weakest link”.

An attacker wishing to make use of a VPN connection will typically not attempt to crack the VPN encryption, since this requires enormous amounts of computation and time. Rather, he/she will see VPN traffic as an indication that there is something really soft and chewy on the other end of the connection. Typically, mobile clients and branch offices are far more attractive targets than the main corporate networks. Once inside those, getting to the corporate network becomes a much easier task.

In designing a VPN, there are many non-obvious issues that need to be addressed. This includes:

- Protecting mobile and home computers.
- Restricting access through the VPN to needed services only, since mobile computers are vulnerable.
- Creating DMZs for services that need to be shared with other companies through VPNs.
- Adapting VPN access policies for different groups of users.
- Creating key distribution policies.

A common misconception is that VPN-connections are equivalents to the internal network from a security standpoint and that they can be connected directly with no further precautions.

It is important to remember that although the VPN-connection itself may be secure, the total level of security is only as high as the security of the tunnel endpoints.

It is becoming increasingly common for users on the move to connect directly to their company's network via VPN from their laptops. However, the laptop itself is often not protected. In other words, an intruder can gain access to the protected network through an unprotected laptop and already-opened VPN connections.

In conclusion, a VPN connection should never be regarded as an integral part of a protected network. The VPN gateways should instead be located in a special DMZ or outside a firewall dedicated to this task. By doing this, you can restrict which services can be accessed via VPN and modem and ensure that these services are well protected against intruders.

In instances where the firewall features an integrated VPN gateway, it is usually possible to dictate the types of communication permitted. The D-Link VPN module features just such a facility.

21.1.1 End Point Security

A basic precaution to take in protecting your network against modem and VPN connection attacks is to ensure that roaming users never communicate directly with the Internet. Instead, they should always be routed through the VPN or modem connection and the company's network, irrespective of whom they wish to communicate with. This way, they enjoy more or less the same level of protection as the rest of the network. For VPN connections, a competent *VPN client* that can block all inbound Internet traffic, aside from that which passes through the VPN connection, must be installed on each portable or home computer.

It is also important to remember that the same restrictions placed on in-house computers should also be placed on the portable or home computers accessing the corporate network. Actually, higher restrictions should be placed on the roaming clients.

End Point Security for Company-owned Computers

Important points that are often included in remote access policies include:

- Anti-virus software is needed to be installed and updated through the remote connection.
- Choose a multi-user operating system where the end user's capabilities may be restricted.
- Do NOT set the VPN/dialup client to automatically remember shared secrets, dialup passwords, or certificates, unless access to such data is password protected using strong encryption.

Any vendor claiming to be capable of securing such data without the user entering a password, using a smart card, or supplying any sort of information, is not telling the truth.

- If the VPN client offers a method for remembering all passwords without having the user supply any information, disable that feature. If not, sooner or later, someone will check that checkbox, and if/when the portable computer is stolen, the thief has an open access route to the corporate network.
- Apply and enforce the same policies as the in-house computers. Such policies usually include:
 - No software downloads from the Internet
 - No games
 - No lending the computer to friends and others
- Schedule inspections of all portable/home computers to verify compliance with all of the above. This process can usually be automated to great extent and even carried out across the remote connection. A few simple script files will usually do to see that no additional software is installed and that registry keys containing values for remembering passwords etc have not been changed.
- Keep data stored locally on portable computers to a minimum to reduce the impact of theft. This includes e-mail cache folders. Actually, it may be best if mail is read through a web gateway, since that leaves the least amount of files in local storage.

- If the above requirements cannot be met, for instance, in cases where the home computer belongs to the employee, then do not grant VPN access.

End Point Security for Partners and other Companies

This subject is usually far more sensitive than securing computers that are actually owned by the company. In cases where management has dictated that a VPN should be established with a partner, subsidiary, or subcontractor that has far more lax security policies, it can become a real nightmare for the IT staff.

It is far from uncommon for a motivated intruder to research companies likely to have connections to his/her target, virtual or otherwise. Should the target's security be too high, it may prove to be far more fruitful to probe other locations that may be used to launch an attack around the primary defense perimeters.

In cases where the security of the remote network cannot be guaranteed, technically and/or physically, it may be a good idea to move shared resources to servers in a separate DMZ and grant remote access only to those servers.

21.1.2 Key Distribution

Plan your key distribution schemes ahead of time. Issues that need to be addressed include:

- By what means to distribute the keys ? Email is not a good idea. Phone conversations might be secure enough. This depends on your local security policy.
- How many different keys should be used? One key per user? One key per group of users? One key per LAN-to-LAN connection? One key for all users and one key for all LAN-to-LAN connections? You are probably better off using more keys than you think necessary today, since it becomes easier to adjust access per user (group) in the future.
- Should the keys be changed? If so, how often? In cases where keys are shared by multiple users, you may want to consider overlapping schemes, so that the old keys work for a short period of time when new keys have been issued.

- What happens when an employee in possession of a key leaves the company? If several users are using the same key, it should be changed of course.
- In cases where the key is not directly programmed into a network unit such as a VPN gateway, how should the key be stored? Should it be on a floppy? As a pass phrase to memorize? On a smart card? If it is a physical token, how should it be handled?

CHAPTER 22

VPN Protocols & Tunnels

22.1 IPsec

IPsec, *Internet Protocol Security*, is a set of protocols defined by the IETF (Internet Engineering Task Force) to provide IP security at the network layer. It is the most widely used standard for implementing VPNs.

IPsec is designed to work for all IP traffic, independently of application. This approach results in the advantage that neither the applications nor the users need to know any details about the encryption.

IPsec uses *Diffie-Hellman key exchange protocol* and *asymmetric encryption* to establish identities, preferred algorithms, and a symmetric key. Then, the algorithm uses the symmetric encryption scheme to encrypt data as it is transferred.

Before IPsec can begin encrypting and transferring the data stream, some preliminary negotiation is necessary. This is accomplished with the *Internet Key Exchange Protocol* (IKE).

In summary, an IPsec based VPN, such as D-Link VPN, is made up by two parts:

- Internet Key Exchange protocol (IKE)
- IPsec protocols (AH/ESP/both)

The first part, *IKE*, is the initial negotiation phase, where the two VPN endpoints agree on which methods will be used to provide security for the underlying IP traffic. Furthermore, IKE is used to manage connections, by defining a set of *Security Associations*, SAs, for each connection. SAs are unidirectional, so there will be at least two SAs per connection.

The second part is the actual IP data transfer, using the *encryption* and *authentication* methods agreed upon in the IKE negotiation. This can be accomplished in a number of ways; by using IPsec protocols ESP, AH, or a combination of both.

The operation flow can be briefly described as follows:

- IKE negotiates how IKE should be protected
- IKE negotiates how IPsec should be protected
- IPsec moves data in the VPN

22.1.1 IPsec protocols

Two primary types of IPsec protocols exist: the *Encapsulating Security Payload (ESP)* protocol and the *Authentication Header (AH)* protocol.

ESP

ESP provides both authentication and encryption to data packets.

AH

AH provides only authentication but not encryption to data packets.

AH does not offer confidentiality to the data transfer and is rarely used; it is *NOT* supported by D-Link firewalls.

Whether IPsec protocol modifies the original IP header or not depends on the IPsec modes.

22.1.2 IPsec Modes

IPsec supports two different modes: *Transport* and *Tunnel* modes.

Transport mode – encapsulates the data of the packet and leaves the IP header unchanged, which is typically used in a client-to-gateway scenario.

Tunnel mode – encapsulates the IP header and payload into a new IPsec packet for transfer, which is typically used in the IPsec gateway-to-gateway scenario.

In transport mode, the ESP protocol inserts an ESP header after the original IP header, and in tunnel mode, the ESP header is inserted after a new outer IP header, but before the original, inner, IP header. All data after the ESP header is encrypted and/or authenticated.

22.1.3 IKE

Encrypting and authenticating data is fairly straightforward, the only things needed are encryption and authentication algorithms, and the keys used with them. The Internet Key Exchange protocol, IKE, is used as a method of distributing these "session keys", as well as providing a way for the VPN endpoints to agree on how the data should be protected.

IKE has three main tasks:

- Provide a means for the endpoints to authenticate each other
- Establish new IPsec connections (create SA pairs)
- Manage existing connections

IKE keeps track of connections by assigning a bundle of *Security Associations*, SAs, to each connection. An SA describes all parameters associated with a particular connection, including things like the *IPsec protocol* used (ESP/AH/both), the *session keys* used to encrypt/decrypt and/or authenticate the transmitted data. An SA is, by nature, unidirectional, thus the need for more than one SA per connection. In most cases, where only one of ESP or AH is used, two SAs will be created for each connection, one describing the incoming traffic, and the other the outgoing. In cases where ESP and AH are used in conjunction, four SAs will be created.

IKE Negotiation

The process of negotiating connection parameters mainly consists of two phases:

IKE Phase-1

- Negotiate how IKE should be protected for further negotiations.

- Authenticate the communication parties, either with *pre-shared key (PSK)* or *certificate*.
- Exchange keying materials with Diffie-Hellman method.
- IKE SAs are created.

IKE Phase-2

– Negotiate how IPsec should be protected.

- Create a pair of IPsec SAs using the IKE SAs from phase-1, detailing the parameters for the IPsec connection.
- Extract new keying material from the Diffie-Hellman key exchange in phase-1, to provide session keys to use in protecting the VPN data flow.

Both the *IKE SAs* and the *IPsec SAs* have limited lifetimes, described as time (seconds), and data (kilobytes). These lifetimes prevent a connection from being used too long, which is desirable from a cryptanalysis perspective.

The IKE phase-1 involves very heavy computation, thus its lifetime is generally longer than the phase-2 IPsec lifetime. This allows for the IPsec connection to be re-keyed simply by performing another phase-2 negotiation. There is no need to do another phase-1 negotiation until the IKE SAs lifetime has expired.

Negotiation Modes

The IKE negotiation has two modes of operation, *main mode* and *aggressive mode*.

The difference between these two is that aggressive mode will pass more information in fewer packets, with the benefit of slightly faster connection establishment, at the cost of transmitting the identities of the security gateways in the clear.

When using aggressive mode, some configuration parameters, such as Diffie-Hellman groups, can not be negotiated, resulting in a greater importance of having "compatible" configurations on both communication ends.

IKE & IPsec Algorithms

There are a number of algorithms used in the negotiation processes. Learning what these algorithms do is essential before attempting to configure the VPN endpoints, since it is of great importance that both endpoints are able to agree on all of these configurations.

IKE & IPsec Encryption

The data flow transferred in VPN connections are encrypted using *symmetric encryption* scheme.

As it is described in [20.2.1 Symmetric Encryption](#), D-Link firewalls support the algorithms listed below:

- DES
- 3DES
- Blowfish
- Twofish
- CAST-128
- AES

DES is only included to be interoperable with some older VPN implementations. Use of DES should be avoided whenever possible, since it is an old algorithm that is no longer considered secure.

Perfect Forward Secrecy (PFS)

Perfect Forward Secrecy (PFS) is an optional property of IKE negotiations. When PFS is configured, the keys that protect data transmission are not used to derive additional keys, and the keying material used to create data transmission keys are not reused.

PFS can be used in two modes, the first is *PFS on keys*, where a new key exchange will be performed in every phase-2 negotiation, that is, a Diffie-Hellman exchange for each IPsec SA negotiation. The other type is *PFS on identities*, where the identities are also protected, by deleting the phase-1 SAs every time a phase-2 negotiation has been finished, making sure no more than one phase-2 negotiation is encrypted using the same key. IKE creates a new SA for every new IPsec SA needed.

PFS is very resource and time consuming and is generally disabled, since it is very unlikely that any encryption or authentication keys will be compromised.

Key Exchange

IKE exchanges the symmetric encryption key using *Diffie-Hellman key exchange protocol*. The level of security it offers is configurable by specifying the *Diffie-Hellman(DH) group*.

The Diffie-Hellman groups supported by D-Link VPN are:

- DH group 1 (768-bit)
- DH group 2 (1024-bit)
- DH group 5 (1536-bit)

The security of the key exchanges increases as the DH groups grow larger, as does the time of the exchanges.

NAT Traversal

One big problem encountered by the IKE and IPsec protocols is the use of NAT, since the IKE and IPsec protocols were not designed to work through NATed network. Because of this, something called "*NAT traversal*" has evolved. NAT traversal is an add-on to the IKE and IPsec protocols that makes them work when being NATed.

In short, NAT traversal is divided into two parts:

- Additions to IKE that lets IPsec peers tell each other that they support NAT traversal, and the specific versions of the draft they support.
- Changes to the ESP encapsulation. If NAT traversal is used, ESP is encapsulated in UDP, which allows for more flexible NATing.

NAT traversal is only used if both ends has support for it. For this purpose, NAT traversal aware VPNs send out a special "vendor ID", telling the other end that it understand NAT traversal, and which specific versions of the draft it supports.

To detect the necessity of using NAT traversal, both IPsec peers send hashes of their own IP addresses along with the source UDP port used in the IKE negotiations. This information is used to see whether the IP

address and source port each peer uses is the same as what the other peer sees. If the source address and port have not changed, then the traffic has not been NATed along the way, and NAT traversal is not necessary. If the source address and/or port has changed, then the traffic has been NATed, and NAT traversal is used.

Once the IPsec peers have decided that NAT traversal is necessary, the IKE negotiation is moved away from UDP port 500 to port 4500. This is necessary since certain NAT devices treat UDP packet to port 500 differently from other UDP packets in an effort to work around the NAT problems with IKE. The problem is that this special handling of IKE packets may in fact break the IKE negotiations, which is why the UDP port used by IKE has changed.

Another problem NAT traversal resolves is regarding the ESP protocol. ESP protocol is an IP protocol and there is no port information like in TCP and UDP, which makes it impossible to have more than one NATed client connected to the same remote gateway at the same time. To solve this problem, ESP packets are encapsulated into UDP. The ESP-UDP traffic is sent on port 4500, the same port as IKE when NAT traversal is used. Once the port has been changed, all following IKE communications are done over port 4500. *Keep-alive* packets are also being sent periodically to keep the NAT mapping alive.

22.1.4 IKE Integrity & Authentication

In the IKE negotiation phase, the authentication to the communicating endpoints is carried out before any actual data transfer, and the integrity of the negotiated message must be secured by sound mathematical algorithms. D-Link VPNs embed various methods for achieving these critical tasks, i.e., hash functions for message integrity, pre-shared keys and X.509 certificates based on asymmetric encryption algorithms (i.e. RSA, DSA) for verifying identities.

Hashing for Integrity

To ensure the message integrity during the IKE negotiation, some hash functions are used by D-Link firewalls to provide message digests for different methods of authentication. The hashing mechanisms ensure that the unchanged messages arrive at the other end after transmission.

D-Link firewalls feature the following two hash functions:

- *SHA-1* – 160-bit message digest.
- *MD5* – 128-bit message digest, faster than SHA-1 but less secure.

Pre-Shared Key (PSK)

Pre-Shared Keys is one of the two primary authentication methods supported by D-Link VPNs. With pre-shared key authentication, an identical symmetric key must be manually configured on both systems. The shared key is a secret passphrase, normally a string of *ASCII characters* or a set of *random Hexadecimal numbers*. In D-Link VPNs, the user can either enter an ASCII password or use the automatic random key generation. Both endpoints need to have the same key defined and the key must be kept secret.

The pre-shared key is used only for the primary authentication; the two negotiating entities then generate dynamic shared session keys for the IKE SAs and IPSec SAs.

The advantages of using PSK are: first, pre-shared keys do not require a central Certificate Authority (CA) or CAs for authentication tasks; second, it provides a means of primary endpoints authentication, based on what, the further IKE negotiation for dynamic session keys can be implemented. The session keys will be used for a limited period of time, where after a new set of session keys are used.

However, one thing that has to be considered when using PSK is the key distribution. How are the pre-shared keys distributed to remote VPN clients and gateways? This is a major issue, since the security of a PSK system is based on the PSKs being secret. Should one PSK be compromised in some way, the configuration will need to be changed to use a new PSK.

X.509 Certificate

The other option for primary authentication is to use *X.509 Certificate* within each VPN gateway. To prove the identity, each gateway owns a certificate signed by a trusted CA. The certificate proves that the public key attached to it truly belongs to the gateway holder, and every gateway also keeps a copy of CA's public key to be able to trust the CA and validate the certificates of other gateways issued from that CA.

Compared to the use of PSK, certificates are more flexible. Many VPN clients, for instance, can be managed without having the same pre-shared key configured on all of them, which is often the case when using pre-shared keys and roaming clients. Instead, should a client be compromised, the client's certificate can simply be revoked. No need to reconfigure every client. But complexity is also added by this method. Certificate-based authentication may be used as part of a larger infrastructure, making all VPN clients and gateways dependent on third parties. In other words, there are more things that have to be configured, and there are more things that can go wrong.

Identification Lists (ID Lists)

When X.509 certificates are used as authentication method, the firewall will accept all remote gateways or VPN clients that are capable of presenting a certificate signed by any of the trusted Certificate Authorities(CAs). This can be a potential problem, especially when using roaming clients.

Consider a scenario where employees on the road shall be given access to the internal corporate networks using VPN clients. The organization administers their own CA, and certificates have been issued to the employees. Different groups of employees are likely to have access to different parts of the internal networks. For instance, members of the sales force need access to servers running the order system, while technical engineers need access to technical databases.

As the IP addresses of the travelling employees VPN clients cannot be foreseen, the incoming VPN connections from the clients cannot be differentiated. This means that the firewall is unable to control the access to various parts of the internal networks.

The concept of *Identification Lists(ID Lists)* presents a solution to this problem. An identification list contains one or more configurable

identities(IDs), where each identity corresponds to the subject field in an X.509 certificate. Identification lists can thus be used to regulate what X.509 certificates are given access to what IPSec connections.

LDAP

LDAP, short for *Lightweight Directory Access Protocol*, is a set of protocols for accessing and downloading information directories. LDAP supports TCP/IP, which is necessary for any type of Internet access. It is used for various applications running on different computer platforms to obtain information from a LDAP server, such as downloading the certificate and CRL registry. The LDAP server maintains the *Certification Authority certificate*, the *Certificate Revocation List(CRL)*, and the end users certificates. The address of the LDAP server can be configured at each VPN endpoint.

IKE XAuth

IKE Extended Authentication (XAuth), is an extended feature to enhance the standard IKE authentication.

XAuth does not replace IKE; it occurs after IKE negotiation phase-1, but before IKE IPSec SA negotiation phase-2. Before XAuth, IKE only supported authentication of the device, not authentication of the user that using the device. With XAuth, IKE can now authenticate the users after the device has been authenticated during phase-1 negotiation. If enabled, a combination of *username & password* will be requested for the add-on user authentication.

22.1.5 Scenarios: IPsec Configuration



Example: Configuring a LAN-to-LAN IPsec Tunnel

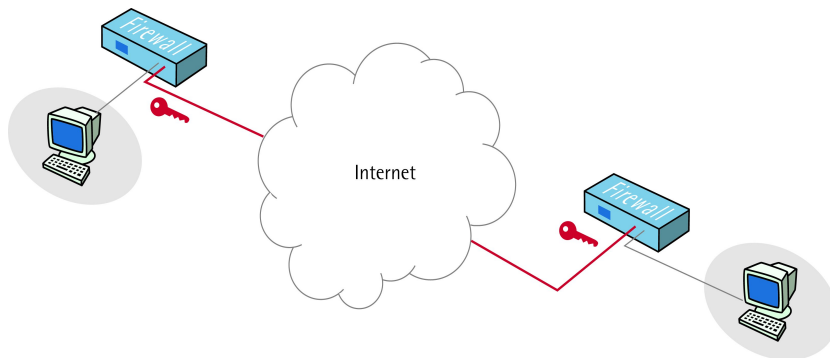


Figure 22.1: LAN-to-LAN Example Scenario.

This example describes how to configure a LAN-to-LAN IPsec tunnel, used to connect a branch office to the head office network.

The head office network use the 10.0.1.0/24 network span with external firewall IP `ip_head_wan`. The branch office use the 10.0.2.0/24 network span with external firewall IP `ip_branch_wan`.

The following configuration will have to be done on both the head office firewall and the branch office firewall.

WebUI :

1. Pre-Shared Key

First of all we need to create a pre-shared key to use for the IPsec authentication.

Objects → **VPN Objects** → **Pre-Shared Keys** → **Add** → **Pre-Shared Key**:

Enter the following:

Name: Enter a name for the pre-shared key, TestKey for instance.

Passphrase/Shared Secret: Enter a secret passphrase.

Passphrase/Confirm Secret: Enter the secret passphrase again.

Then click **OK**

2. IPsec Tunnel

Next step is to configure the IPsec tunnel.

General

Interfaces → **IPsec Tunnels** → **Add** → **IPsec Tunnel**:

Enter the following:

Name: IPsecTunnel

Local Network: This is the local network that the remote users will connect to. So in the head office firewall 10.0.1.0/24 will be used and in the branch office firewall 10.0.2.0/24 will be used.

Remote Network: This is the network that the remote users will connect from. So in the head office firewall 10.0.2.0/24 will be used and in the branch office firewall 10.0.1.0/24 will be used.

Remote Endpoint: This is the public ip's of each firewall, where the tunnels will be terminated. This means that the head office firewall will use ip_branch_wan and the branch office firewall will use ip_head_wan.

Encapsulation Mode: Tunnel

Algorithms

IKE Algorithms: Medium or High

IPsec Algorithms: Medium or High

Authentication

Pre-Shared Key: Select the pre-shared key created earlier, TestKey in this case.

Then click **OK**

3. Configure Route

Next step is to configure the route to the IPsec tunnel.

Routing → **Main Routing Table** → **Add** → **Route**:

Enter the following:

Interface: IPsecTunnel

Network: On the head office firewall 10.0.2.0/24 and on the branch office firewall 10.0.1.0/24.

Then click **OK**

4. Configure Rules

Finally we need to configure the rules to allow traffic inside the tunnel. See [14.3 IP Rules Configuration](#) for details on how to configure rules.



Example: Configuring a IPsec Tunnel for Roaming Clients

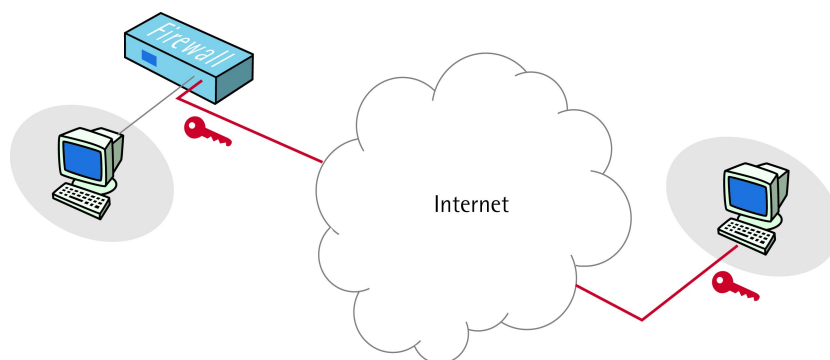


Figure 22.2: IPsec Roaming Client Example Scenario.

This example describes how to configure a IPsec tunnel, used for roaming clients (mobile users) that connect to the head office to gain remote access. The head office network use the 10.0.1.0/24 network span with external firewall IP ip_wan.

The following configuration will have to be done on the head office firewall.

WebUI :

1. Pre-Shared Key

First of all we need to create a pre-shared key to use for the IPsec authentication.

Objects → **VPN Objects** → **Pre-Shared Keys** → **Add** → **Pre-Shared Key**:

Enter the following:

Name: Enter a name for the pre-shared key, SecretKey for instance.

Passphrase/Shared Secret: Enter a secret passphrase.

Passphrase/Confirm Secret: Enter the secret passphrase again.

Then click **OK**

2. IPsec Tunnel

Next step is to configure the IPsec tunnel.

General

Interfaces → **IPsec Tunnels** → **Add** → **IPsec Tunnel**:

Enter the following:

Name: RoamingIPsecTunnel

Local Network: 10.0.1.0/24 (This is the local network that the roaming users will connect to)

Remote Network: The firewall looks at this field and compares it to the roaming user's source IP address in order to allow connections only from the configured local net to remote net. However, in this scenario, clients should be allowed to roam in from everywhere. Thus, this field is set to all-nets (0.0.0.0/0). That means that virtually all existing IPv4-addresses are allowed to connect.

Remote Endpoint: (None)

Encapsulation Mode: Tunnel

Algorithms

IKE Algorithms: Medium or High

IPsec Algorithms: Medium or High

Authentication

Pre-Shared Key: Select the pre-shared key created earlier, SecretKey in this case.

Routing

Automatic Routing

The IPsec tunnel needs to be configured to dynamically add routes to the remote network when the tunnel is established. This is done under the Routing tab. **Dynamically add route to the remote network when a tunnel is established:** Enable

Then click **OK**

3. Configure Rules

Finally we need to configure the rules to allow traffic inside the tunnel. See [14.3 IP Rules Configuration](#) for details on how to configure rules.

22.2 PPTP/ L2TP

As introduced in the previous sections, IPsec provides methods for two endpoints to transport data packets as they are connecting by a "private channel". Such technique is often called *Tunneling*. Like the functions of IPsec we have discussed, the tunneling protocols offer the standards for encapsulation, transmission, and decapsulation to the data transfer process. The endpoints of the tunnel must agree on the *same* tunneling protocol to be able to communicate.

IPsec features the *Tunnel mode ESP* encapsulation with *encryption* and *authentication* and becomes widely used for very secure VPN implementations. However, there are some limitations of using IPsec tunneling, for example, it is not supported by all systems and it can be hard to configure.

In contrast, PPTP and L2TP tunneling protocols are widely supported and easier to configure than IPsec.

22.2.1 PPTP

Point-to-Point Tunneling Protocol (PPTP) is built on Point-to-Point protocol (PPP), *Generic Routing Encapsulation (GRE)*, and TCP/IP.

PPTP tunneling format

PPTP relies on the PPP protocol to encapsulate datagrams (see [9.4.1 PPP](#)). The PPP frame is then encapsulated into GRE packet with routing information included, which is in turn packed with an IP header to conform to the Internet addressing convention, shown in [Table 22.1](#). The Layer 2 data frame is the basic transport unit. Data-link layer header and trailer are put onto the PPTP encapsulated packet to form the tunneling data. PPTP uses TCP port 1723 for its control connection and GRE (IP protocol 47) for the PPP data.

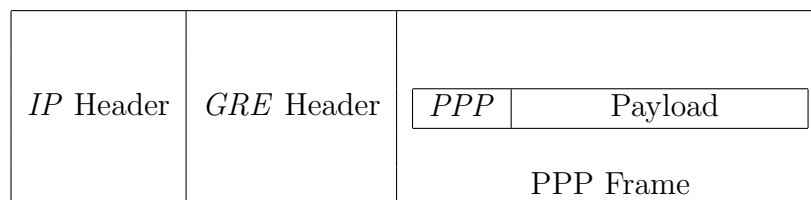


Table 22.1: PPTP Encapsulation.

PPTP authentication

Authentication as an option in PPTP is derived directly from PPP, such as:

- Password Authentication Protocol (PAP)
- Challenge Handshake Authentication Protocol (CHAP)
- Microsoft CHAP version 1 and version 2

PAP is a plaintext authentication scheme by requesting and sending user name and password in plaintext. Therefore it is not resistant to Man-in-the-middle attack and dictionary attack as the remote access client's password can be easily intercepted and determined. Moreover, PAP offers NO protection against replay attacks and spoofing.

CHAP uses MD5 algorithm to hash a challenge and protects against replay attacks by using an arbitrary challenge string per authentication attempt. This is better than PAP since the password is never sent over the link. Instead, the password is used to create the one-way MD5 hash. That means that CHAP requires passwords to be stored in a *reversibly encrypted* form.

MS-CHAP v1 is similar to CHAP, the main difference is that with MS-CHAP v1 the password only needs to be stored as a MD4 hash instead of a reversibly encrypted form.

MS-CHAP v2 is similar to MS-CHAP v1 with the difference that the server also authenticates itself with the client.

PPTP encryption

Initially, PPP connection does not use encryption. To provide confidentiality to the tunneling, the *Microsoft Point-to-Point Encryption (MPPE)* may be used with PPTP to support an encrypted data tunnel.

MPPE uses the RSA RC4 algorithm for encryption and supports 40-bit, 56-bit and 128-bit session keys, which are changed frequently to ensure security. However, the initial encryption key is derived based on user's password, and hence it may be vulnerable to attacks.

Since PPTP security is password-based, the choice of a good password is an important security consideration. Regardless of the key length chosen (40, 56 or 128-bit), the true strength of the key is governed by the randomness of the password.



Example: Configuring PPTP Server

This example describes how to set up a PPTP server. The LAN network is a 192.68.1.0/24 network, and 10.0.0.0/24 is the network on the WAN interface. PPTP clients will connect to the PPTP server on 10.0.0.1 on the WAN interface, in order to access resources on the LAN interface.

WebUI :

1. Local User Database

We need to create a local user database to store user information in. For more information, see [17.2.1 Local User Database](#) section.

User Authentication → **VPN Local User Databases** → **Add** → **Local User Database**:

Enter a name for the user database, "UserDB" will be used in this example:

Name: UserDB

Then click **OK**

2. Add User to Local User Database

We need to add a user to the local user database we created above.

User Authentication → **VPN Local User Databases** → **UserDB** → **Add** → **User**:

Enter the following:

Username: testuser

Password: testpassword

Confirm Password: testpassword

It is possible to configure a static IP for this user in the Per-user PPTP/L2TP IP Configuration section. Then click **OK**

3. PPTP Server

Next step is to configure the PPTP server.

Interfaces → **L2TP/PPTP Servers** → **Add** → **L2TP/PPTP Server**:

Enter the following:

Name: PPTPServer

Inner IP Address: This is the IP address of the PPTP server inside the tunnel. In this case 192.168.1.1

Tunnel Protocol: PPTP

Outer Interface Filter: WAN

Server IP: This is the IP that remote users will connect to, in this case 10.0.0.1

Use User Authentication Rules: Enable (Specifies that authentication should be performed)

Microsoft Point-to-Point Encryption (MPPE): Select the encryption strengths to allow.

IP Pool: 192.168.1.10-192.168.1.20 (The pool of IP addresses to assign IP:s from to connecting users)

DNS (Primary/Secondary): Specify any eventual DNS servers to hand out to connected clients.

NBNS (Primary/Secondary): Specify any eventual NBNS (WINS) servers to hand out to connected clients.

Proxy ARP: Leave as default, or specifically select the LAN interface if the IP:s in the IP Pool are a part of the network on the LAN interface.

Then click **OK**

4. User Authentication Rule

Next step is to configure the user authentication rule to use for authentication.

User Authentication → **User Authentication Rules** → **Add** → **User Authentication Rule**:

Enter the following:

Name: PPTPUARule

Agent: PPP

Authentication Source: Local

Interface: L2TPServer

Originator IP: 0.0.0.0/0 (all-nets)

Terminator IP: 10.0.0.1 (The IP the PPTP server is listening on)

Authentication Options/Local User DB: UserDB (The user database created earlier)

PPP Agent Options: Select the authentication protocols to support. (Default setting is to support all authentication protocols)

Then click **OK**

5. IP Rules

The final step is to set up a rule to allow traffic from PPTP clients onto the LAN network.

Rules → **IP Rules** → **Add** → **IP Rule**:

Enter the following:

Name: FromPPTPClients

Action: Allow

Service: Any

Source Interface: PPTPServer

Source Network: 192.168.1.10-192.168.1.20

Destination Interface: LAN

Destination Network: 192.168.1.0/24 (Network on LAN interface)

Then click **OK**

If the PPTP clients should be able to access external resources (such as the Internet for example) a NAT rule has to be configured as well.

When the configuration is saved and activated, it should be possible for PPTP clients to connect to the PPTP server on 10.0.0.1 on the WAN interface.



Example: Configuring PPTP Client

This example describes how to set up a PPTP client. The PPTP server is located at 10.0.0.1 and all traffic should be routed over the PPTP tunnel.

WebUI :

1. PPTP Client

First step is to configure the PPTP client.

Interfaces → **L2TP/PPTP Servers** → **Add** → **L2TP/PPTP Server:**

Enter the following:

Name: PPTPClient

Tunnel Protocol: PPTP

Remote Endpoint: 10.0.0.1 (The IP of the PPTP server)

Remote Network: 0.0.0.0/0 (all-nets, as we will route all traffic into the tunnel)

Username: The username provided to you by your service provider.

Password: The password provided to you by your service provider.

Confirm Password: Retype the password.

We keep the default settings for authentication and encryption. If dial-on-demand is enabled, the tunnel will only be up when there is traffic on the PPTP client interface. It is possible to configure how the firewall should sense activity on the interface, and how long time to wait with no activity before the tunnel is disconnected. Then click **OK**

2. Routes

The final step is to configure a single-host route to the PPTP server over the WAN interface.

Routing → Main Routing Table → Add → Route:

Enter the following:

Interface: WAN

Network: 10.0.0.1 (IP of the PPTP server)

Gateway: The gateway on the WAN network. None if no gateway is used.

Local IP Address: (None)

Metric: 0

Then click **OK**

When the configuration is saved and activated, the PPTP client should connect to the PPTP server, and all traffic (except traffic to 10.0.0.1) should be routed over the PPTP interface.

22.2.2 L2TP

The Layer Two Tunneling Protocol (L2TP) is an extension based on PPP, which is more flexible than PPTP and IPsec in that it uses the UDP protocol for communication, which makes it easier to traverse routers with NAT. In addition, L2TP supports multiple calls for each tunnel while only one connection per tunnel is allowed by PPTP or IPsec tunneling.

L2TP tunneling format

L2TP relies on the PPP protocol to encapsulate datagrams (see [9.4.1 PPP](#)). The PPP frame is then encapsulated into a L2TP header, which is in turn packed with an UDP and IP header to conform to the Internet addressing convention, shown in [Table 22.2](#). Data-link layer header and trailer are put onto the L2TP encapsulated packet to form the tunneling data. L2TP uses UDP port 1701 for its control and data connections.

L2TP authentication

PPTP and L2TP tunnels use the same authentication mechanisms as PPP connections such as, PAP, CHAP, MS-CHAP v1 and v2.

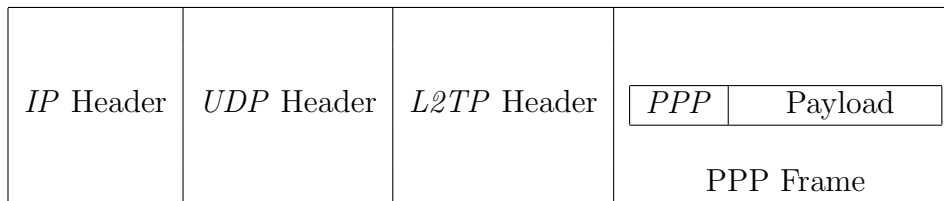


Table 22.2: L2TP Encapsulation.

L2TP encryption

L2TP calls for MPPE for encryption.

L2TP/IPsec

The authentication methods addressed by PPTP and L2TP mainly rely on the user's password, and the encryption to the tunneling data is not initially designed for these protocols. Thus, PPTP and L2TP are NOT resistant to many common attacks, e.g. *Man-in-the-middle*, *Replay*, *Spoofing*, *Dictionary*, and *Dos* attacks.

L2TP and IPsec can work together to benefit from both flexibility and stronger security. By encapsulating L2TP as payload into an IPsec packet, connections can be protected by sound encryption and authentication. This combination is called *L2TP/IPsec*.

How To: Configure L2TP in D-Link Firewall

In this section, guidelines and examples for configuring L2TP clients and servers are covered.

**Example: Configuring L2TP/IPsec Server (PSK)**

This example describes how to set up a L2TP server with IPsec, using pre-shared keys. The LAN network is a 192.68.1.0/24 network, and 10.0.0.0/24 is the network on the WAN interface. L2TP clients will connect to the L2TP/IPsec server on 10.0.0.1 on the WAN interface, in order to access resources on the LAN interface.

WebUI :

1. Pre-Shared Key

First of all we need to create a pre-shared key to use for the IPsec authentication.

Objects → **VPN Objects** → **Pre-Shared Keys** → **Add** → **Pre-Shared Key**:

Enter the following:

Name: Enter a name for the pre-shared key, L2TPKey for instance.

Passphrase/Shared Secret: Enter a secret passphrase.

Passphrase/Confirm Secret: Enter the secret passphrase again.

Then click **OK**

2. Local User Database

We need to create a local user database to store user information in. For more information, see [17.2.1 Local User Database](#) section.

User Authentication → **VPN Local User Databases** → **Add** → **Local User Database**:

Enter a name for the user database, "UserDB" will be used in this example:

Name: UserDB

Then click **OK**

3. Add User to Local User Database

We need to add a user to the local user database we created above.

User Authentication → **VPN Local User Databases** → **UserDB** → **Add** → **User**:

Enter the following:

Username: testuser

Password: testpassword

Confirm Password: testpassword

It is possible to configure a static IP for this user in the Per-user PPTP/L2TP IP Configuration section. Then click **OK**

4. IPsec Tunnel

Next step is to configure the IPsec tunnel.

General

Interfaces → **IPsec Tunnels** → **Add** → **IPsec Tunnel**:

Enter the following:

Name: L2TPIPsecTunnel

Local Network: This is the local network that the remote users will connect to. As we are going to use L2TP this is the IP the L2TP clients will connect to. In this case 10.0.0.1

Remote Network: The firewall looks at this field and compares it to the roaming user's source IP address in order to allow connections only from the configured local net to remote net. However, in this scenario, clients should be allowed to roam in from everywhere. Thus, this field is set to all-nets (0.0.0.0/0). That means that virtually all existing IPv4-addresses are allowed to connect.

Remote Endpoint: (None)

Encapsulation Mode: Transport

Algorithms

IKE Algorithms: Medium

IPsec Algorithms: Medium

Authentication

Pre-Shared Key: Select the pre-shared key created earlier, L2TPKey in this case.

Automatic Routing

The IPSec tunnel needs to be configured to dynamically add routes to the remote network when the tunnel is established. This is done under the Routing tab. **Dynamically add route to the remote network when a tunnel is established:** Enable

Then click **OK**

5. L2TP Server

Next step is to configure the L2TP server.

Interfaces → L2TP/PPTP Servers → Add → L2TP/PPTP Server:

Enter the following:

Name: L2TPServer

Inner IP Address: This is the IP address of the L2TP server inside the tunnel. In this case 192.168.1.1

Tunnel Protocol: L2TP

Outer Interface Filter: L2TPIPsecTunnel

Server IP: This is the IP that remote users will connect to, in this case 10.0.0.1

Use User Authentication Rules: Enable (Specifies that authentication should be performed)

Microsoft Point-to-Point Encryption (MPPE): Select the encryption strengths to allow.

IP Pool: 192.168.1.10-192.168.1.20 (The pool of IP addresses to assign IP:s from to connecting users)

DNS (Primary/Secondary): Specify any eventual DNS servers to hand out to connected clients.

NBNS (Primary/Secondary): Specify any eventual NBNS (WINS) servers to hand out to connected clients.

Proxy ARP: Leave as default, or specifically select the LAN interface if the IP:s in the IP Pool are a part of the network on the LAN interface.

Then click **OK**

6. User Authentication Rule

Next step is to configure the user authentication rule to use for authentication.

User Authentication → **User Authentication Rules** → **Add** → **User Authentication Rule:**

Enter the following:

Name: L2TPUARule

Agent: PPP

Authentication Source: Local

Interface: L2TPServer

Originator IP: 0.0.0.0/0 (all-nets)

Terminator IP: 10.0.0.1 (The IP the L2TP server is listening on)

Authentication Options/Local User DB: UserDB (The user database created earlier)

PPP Agent Options: Select the authentication protocols to support. (Default setting is to support all authentication protocols)

Then click **OK**

7. IP Rules

The final step is to set up a rule to allow traffic from L2TP clients onto the LAN network.

Rules → **IP Rules** → **Add** → **IP Rule:**

Enter the following:

Name: FromL2TPClients

Action: Allow

Service: Any

Source Interface: L2TPServer

Source Network: 192.168.1.10-192.168.1.20

Destination Interface: LAN

Destination Network: 192.168.1.0/24 (Network on LAN interface)

Then click **OK**

If the L2TP clients should be able to access external resources (such as the Internet for example) a NAT rule has to be configured as well.

When the configuration is saved and activated, it should be possible for L2TP/IPsec clients to connect to the L2TP/IPsec server on 10.0.0.1 on the WAN interface.



Example: Configuring L2TP/IPsec Client

This example describes how to set up a L2TP client with IPsec, using pre-shared keys. The L2TP server is located at 10.0.0.1 and all traffic should be routed over the L2TP tunnel.

WebUI :

1. Pre-Shared Key

First of all we need to create a pre-shared key to use for the IPsec authentication.

Objects → **VPN Objects** → **Pre-Shared Keys** → **Add** → **Pre-Shared Key**:

Enter the following:

Name: Enter a name for the pre-shared key, L2TPKey for instance.

Passphrase/Shared Secret: Enter the secret passphrase. (Has to be the same as configured on the L2TP/IPsec server)

Passphrase/Confirm Secret: Enter the secret passphrase again.

Then click **OK**

2. IPsec Tunnel

Next step is to configure the IPsec tunnel.

General

Interfaces → IPsec Tunnels → Add → IPsec Tunnel:

Enter the following:

Name: L2TPIPsecTunnel

Local Network: IP of the interface to connect from.

Remote Network: 10.0.0.1 (As this is where the L2TP/IPsec server is located)

Remote Endpoint: (None)

Encapsulation Mode: Transport

Algorithms

IKE Algorithms: Medium

IPsec Algorithms: Medium

Authentication

Pre-Shared Key: Select the pre-shared key created earlier, L2TPKey in this case.

Automatic Routing

The IPSec tunnel needs to be configured to **not** dynamically add routes to the remote network when the tunnel is established. This is done under the Routing tab. **Dynamically add route to the remote network when a tunnel is established:** Disable

Then click **OK**

3. L2TP Client

Next step is to configure the L2TP client.

Interfaces → L2TP/PPTP Servers → Add → L2TP/PPTP Server:

Enter the following:

Name: L2TPClient

Tunnel Protocol: L2TP

Remote Endpoint: 10.0.0.1 (The IP of the L2TP/IPsec server)

Remote Network: 0.0.0.0/0 (all-nets, as we will route all traffic into the tunnel)

Username: The username provided to you by your service provider.

Password: The password provided to you by your service provider.

Confirm Password: Retype the password.

We keep the default settings for authentication and encryption. If dial-on-demand is enabled, the tunnel will only be up when there is traffic on the L2TP client interface. It is possible to configure how the firewall should sense activity on the interface, and how long time to wait with no activity before the tunnel is disconnected. Then click **OK**

4. Routes

The final step is to configure a single-host route to the L2TP/IPsec server over the IPsec interface.

Routing → Main Routing Table → Add → Route:

Enter the following:

Interface: L2TPIPsecTunnel

Network: 10.0.0.1 (IP of the L2TP/IPsec server)

Gateway: (None)

Local IP Address: (None)

Metric: 0

Then click **OK**

When the configuration is saved and activated, the L2TP/IPsec client should connect to the L2TP/IPsec server, and all traffic (except traffic to 10.0.0.1) should be routed over the L2TP/IPsec interface.

22.3 SSL/TLS (HTTPS)

The *Secure Sockets Layer (SSL)* protocol is a *browser-based* secure transaction standard alternative to IPsec-based VPNs.

It requires little or no software or hardware on remote PCs, and the secure connection is mainly operated by the web browser and the web server, which is a easier implemented and more cost-efficient means compared to the IPsec scheme. Further more, it can easily provide user-by-user authentication.

Built upon private key encryption and public key authentication, SSL provides privacy and data integrity between two communicating applications over TCP/IP. In the OSI module, the SSL protocol layer is placed between the connection-oriented network layer protocol TCP/IP and the application layer(e.g. HTTP). It relies on *certificates* for entity authentication and the entity's public key is used to negotiate the *symmetric key* for traffic encryption.

The *Transport Layer Security (TLS)*, is the successor to SSL and provides much the same functionality but with much firmer standardization and foothold in the IETF.

The HTTP running on top of SSL/TLS is often called HTTPS, which is one common use of SSL/TLS to secure web browsing service between a browser and a web server. When you visit "secure" web sites, you may have noticed that the URLs begin with the letters "https://" rather than "http://". This is HTTP wrapped up inside SSL/TLS. Most commonly used web browsers support HTTPS, and more and more web sites use the protocol to obtain confidential user information, such as credit card numbers.

There are a number of versions of the SSL/TLS protocol. D-Link firewalls fully support *SSLv3* and *TLSv1*.

Part IX

Traffic Management

Traffic management is concerned with controlling and allocating network bandwidth and minimizing possible delay and congestion on networks. It encompasses the measuring of network capacity and traffic modelling to manage network resources efficiently and provide services the bandwidth they need.

Topics in this part includes:

- [Traffic Shaping](#)
- [Server Load Balancing \(SLB\)](#)

CHAPTER 23

Traffic Shaping

23.1 Overview

TCP/IP networks are being called upon to carry traffic belonging to a growing variety of users with diverse service requirements, for example, bulk data transfer, IP Telephony, VPNs, and multimedia applications. But one of the major drawbacks of TCP/IP is the lack of true *Quality of Service (QoS)* functionality, which is the ability to guarantee and limit bandwidth for certain services and users. Although there are protocols like *Diff-Serv (Differentiated Services)* and other solutions that intend to offer QoS in large and complex networks, none of the solutions have reached a high enough standard for large-scale usage.

Another fact is that most of the current QoS solutions are application-based, that is, they work by having applications supplying the network with QoS information. From a security standpoint, it is of course unacceptable that the applications (that is, the users) decide the priority of their own traffic within a network. In security-sensitive scenarios, where the users cannot be trusted, the network equipment should be the sole arbiter of priorities and bandwidth allocations.

To address the above problems, D-Link firewalls provide QoS functionality, and apply limits and guarantees for QoS to the network traffic itself, rather than trusting the applications/users to make the choices. It is hence well suited to manage bandwidth for a small LAN as well as in one or more

choke points in large WANs.

23.1.1 Functions

The simplest way to obtain QoS in a network, seen from a security as well as a functionality perspective, is to have the components in the network, known as traffic shapers, be responsible for network traffic control in well-defined choke points. A D-Link firewall has an extensible traffic shaper integrated inside.

The traffic shaper works by measuring and queuing IP packets, in transit, with respect to a number of configurable parameters. Differentiated rate limits and traffic guarantees based on source, destination, and protocol parameters can be created, much the same way firewall rules are implemented. The main functions can be summarized as follows:

- Applying bandwidth limits by queuing packets that would exceed configured limits into packet buffers, and sending them later when the momentary demand for bandwidth is lower.
- Dropping packets if the packet buffers are full. The packet to be dropped should be chosen from those that are responsible for the congestion.
- Prioritizing traffic according to the administrator's choice; if the traffic in a higher priority increases while a communication line is full, traffic in lower priorities should be temporarily limited to make room for the high-priority traffic.
- Providing bandwidth guarantees. This is typically accomplished by treating a certain amount of traffic (the guaranteed amount) as a higher priority, and traffic exceeding the guarantee as the same priority as "any other traffic", which then gets to compete with the rest of the non-prioritized traffic.

Well-built traffic shapers do not normally work by queuing up immense amounts of data and then sorting out prioritized traffic to send before sending non-prioritized traffic. Rather, they attempt to measure the amount of prioritized traffic and then limit the non-prioritized traffic dynamically, so that it will not interfere with the throughput of prioritized traffic.

23.1.2 Features

The traffic shaper in D-Link firewalls has the following key features:

- **Pipe based**
Traffic shaping in D-Link firewalls is handled by a concept based on "pipes", where each pipe has several prioritizing, limiting and grouping possibilities. Individual pipes may be chained in different ways to construct bandwidth management units that far exceed the capabilities of one single pipe.
- **Traffic prioritizing and bandwidth limiting**
Every pipe contains a number of priority levels, each with its own bandwidth limit, specified in kilobits per second. Limits may also be specified for the total of the pipe.
- **Grouping**
Traffic through a pipe can be automatically grouped into "pipe users", where each pipe user can be configured to the same extent as the main pipe. A group is specified with respect to a number of parameters, for instance, source or destination IP network, IP address or port number.
- **Dynamic bandwidth balancing**
The traffic shaper can be used to dynamically balance the bandwidth allocation of different pipe groups if the pipe as a whole has exceeded its limits. This means that available bandwidth is evenly balanced with respect to the grouping for the pipe.
- **Pipe chaining**
When pipes are assigned to pipe rules, up to eight pipes may be concatenated to form a chain. This permits filtering and limiting to be handled in a very sophisticated manner.
- **Traffic guarantees**
With the proper pipe configuration, the traffic shaping may be used to guarantee bandwidth (and thereby quality) for traffic through the firewall.

A closer look into these features are given in the sections next.

23.2 Pipes

A **Pipe** is a central concept in the traffic shaping functionality of D-Link firewalls and is the base for all bandwidth control. Pipes are fairly

simplistic, in that they do not know much about the types of traffic that pass through them, and they know nothing about the direction either. A pipe simply measures the amount of traffic that passes through it and applies the configured limits in each precedence and/or user group. The task of traffic filtering, categorizing, and prioritizing is done by **Pipe Rules** covered in the next section.

D-Link firewalls are capable of handling hundreds of pipes simultaneously, but in reality, only a handful of pipes are required for most setups. The only occasion that uses dozens of pipes is the scenario where an individual pipe is created for each service (protocol, or client in ISP cases).

23.2.1 Precedences and Guarantees

Depending on particular applications or manual configurations, traffics can be treated as having different levels of importance.

In an IP version 4 packet, there is a 1-byte field called *Type-of-Service(ToS)* in the header (shown in Table 23.1). This ToS field is used in Diff-Serv approach to provide QoS by differentiating classes of service into different priorities to support various network applications. The six left-most bits of this field is called *Differentiated Services Code Point(DSCP)* and the last two bits were not defined within the Diff-Serv model. The Diff-Serv standard utilizes the higher 3 bits of DSCP for application priority setting, which is organized into 8 precedence levels from 0 to 7; and the lower 3 bits are used to offer finer granularity for precedence definitions. The priority of an application increases with 0 the lowest and 7 the highest. The values 6 and 7 are reserved for network control packets, so the values through 0-5 can be set for priority based on IP networks or applications.

Corresponding to these 8 levels, a pipe in a D-Link firewall contains 4 precedences – **Low, Medium, High, and Highest** – for clarifying the relative importance of the traffic. Each of these precedences maps to 2 levels in DSCP definition, for example, "Low" stands for level 0 and 1. Traffic in precedence "Medium" will be passed on before traffic in precedence "Low", traffic in precedence "High" before "Medium" and "Low", and so on. The precedence assignment is controlled by the **Pipe Rules**. In order to determine what precedence the traffic belongs to, each packet buffer is assigned a precedence number before it is sent into a pipe.

The actual limiting of bandwidth is performed inside each precedence;

separate bandwidth limits may be specified for each of the 4 precedences with a unit of "kilobits per second". Traffic that exceeds the limit of a higher precedence will automatically be transferred into the "Low" level for best effort delivery, as long as there is room in this precedence.

1 byte		1 byte	2 bytes	
Version	IP Header Length	<i>Type-of-Service</i>	Total Length	
Identification			Flags	Fragment Offset
Time-to-Live		Protocol	Header Checksum	
Source Address				
Destination Address				
Options(padding)				
Data				

Table 23.1: IPv4 Packet Format

In addition to the limit per precedence, a limit for the pipe as a whole may also be specified. When the bandwidth utilization through the pipe reaches the total limit, traffic will be prioritized depending on what precedence it belongs to. Higher precedences have a greater chance of making it through the pipe without queuing.



■

- 1. Set a total limit**

In order to know how much to limit lower-precedence traffic, the pipe needs to know the overall capability.

- 2. Limits cannot be higher than the available connection bandwidth**

If the pipe limits are set higher than the actual available bandwidth, the pipe will never know that the connection is full, and hence be unable to throttle the lower-precedence traffic.

- 3. Bandwidth cannot be guaranteed if available bandwidth is not known at all times**

For any traffic shaper to work, it needs to know the bandwidth passing through the choke point that it is trying to "protect". If the connection is shared with other users or servers that are not under the control of the firewall, it is nearly impossible to guarantee bandwidth, simply because the firewall will not know how much bandwidth is available for the connection. Simple limits will of course work, but guarantees, priorities and dynamic balancing will not.

4. Watch for leaks

Make sure that all traffic that is desired for bandwidth control passes through the pipes.

■

23.2.2 Grouping Users of a Pipe

If pipes were restricted to the functionality described so far, traffic would be limited without respect to source or destination. This mode of operation is likely sufficient for managing simple traffic limits and guarantees.

However, D-Link firewalls have the ability to group traffic within each pipe. This means that traffic will be classified and grouped with respect to the source or destination of each packet passing through the pipe.

Grouping may be performed on **source/destination network, IP address, port, or interface**. In the network grouping cases, the network size may be specified. The port grouping cases include the IP address, meaning that port 1024 of computer A is not the same "group" as port 1024 of computer B.

The benefit of using grouping is that additional bandwidth controls may be applied to each group. This means that if grouping is performed on, for example, IP address grouping, the firewall can limit and guarantee bandwidth per IP address communicating through the pipe.

Limits can be set either by specifying the maximum bandwidth per group manually or using the Dynamic Balancing. The control first occurs per user group and then continues with the pipe as a whole.

23.2.3 Dynamic Bandwidth Balancing

As previously stated, per-user bandwidth may be limited by enabling grouping within a pipe. This may be used to ensure that one group cannot consume all of the available bandwidth. But what if the bandwidth for the pipe as a whole has a limit, and that limit is exceeded?

Such problem is addressed by a feature in D-Link firewalls called **Dynamic Balancing**. This algorithm ensures that the bandwidth limit of each group is dynamically lowered (or raised) in order to evenly balance the available bandwidth between the groups of the pipe. The temporary restriction will be removed until the configured limit is satisfied.

The dynamic adjustments take place 20 times per second, and will quickly adapt to changed bandwidth distributions.

Dynamic balancing functions within each precedence of a pipe individually. This means that if groups are allotted a certain small amount of high priority traffic, and a larger chunk of best-effort traffic, all groups will get their share of the high-precedence traffic as well as their fair share of the best-effort traffic.

23.3 Pipe Rules

Pipe Rules are policies that make decisions of what traffic should be passed through which pipes. The pipe rule filters the traffic by service type and interface & network IP addresses, much in the same way as the normal IP rules. Then, the rule chooses appropriate forward and return pipes to the traffics, and determines the precedence(priority) on it. When the firewall receives traffics, it will be able to find these pipe and precedence information in matching rules, and control the utilization of bandwidth according to the limits and/or grouping defined in specific pipes. Remember that only traffic matching a pipe rule will be traffic shaped, and the first matching rule is the one used.

23.4 *Scenarios*: Setting up Traffic Shaping

As seen from the previous sections, in D-Link firewalls, all measuring, limiting, guaranteeing and balancing is carried out in **Pipes**. However, a pipe by itself is meaningless unless it is put into use in the **Pipe Rules**

section. Each rule can pass traffic through one or more pipes, in a precedence(priority) of the administrator's choice.

Network traffic is first filtered within the firewall's normal IP ruleset; if allowed, it is then compared with the Pipe Rules section and passed to the pipe(s) specified in the matching pipe rule. In the pipe, traffic is limited with respect to the configuration and is then forwarded to its destination, or to the next pipe in a chain.

To summarize, the following steps are necessary for setting up traffic shaping:

1. **Traffic shaping requirements planning**

If requirements to the current network, such as how traffic should be limited, prioritized, guaranteed, or distributed are unclear, the configuration work will be more confusing than helpful.

2. **Pipes setup**

Set up pipes that describe limits for different precedences, and define grouping criterion.

3. **Pipe rules setting**

Assign, in Pipe Rules, specific type of service, address filter, precedence, and different pipes/chains to use for both forward & return directions.

4. **Verification**

Verify that the configured traffic shaping works in the desired manner.



Example: Applying a basic two-way bandwidth limits

In this example, two pipes for controlling both inbound and outbound traffics are created, named "std-in" and "std-out" respectively, and a total pipe limit of 1000 kilobits per second is set to each of them. This pair of pipes simply limits all traffic that gets passed through each direction to 1000 kbps, regardless of what traffic it is.

After setting the total limits in the two pipes, two pipe rules need to be specified to assign pipes onto proper directions, interfaces, and networks. Since these two primary rules are applied to all possible services, the fixed precedence "Low" is defined on them.

WebUI :

1. Pipes

Pipe "std-in" for inbound traffic:

Enter the following and then click **OK**.

Traffic Shaping → **Pipes** → **Add** → **Pipe**:

General

Name: std-in

Pipe Limits

Total: 1000

Pipe "std-out" for outbound traffic:

Create the other pipe using the same steps as above with the name changed to "std-out"

2. Pipe Rules

Rule "ToInternet" assigning pipes to traffics going through the firewall from LAN to WAN for all services(defined by the Services object "all-services"):

Traffic Shaping → **Pipe Rules** → **Add** → **Pipe Rule**:

→ **General**

Enter the following:

Name: ToInternet

Service: all-services

Address Filter

	Source	Destination
--	---------------	--------------------

Interface:	lan	wan
-------------------	-----	-----

Network:	lanet	wannet
-----------------	-------	--------

→ **Traffic Shaping**

Pipe Chains

Forward Chain: Select "std-out" from **Available** list and put it into **Selected** list.

Return Chain: Select "std-in" from **Available** list and put it into **Selected** list.

Precedence

Check **Use Fixed Precedence**

Select **Low** from the dropdown list and then click **OK**.

Rule "FromInternet" assigning pipes to traffics going through the firewall from WAN to LAN for "all-services":

Traffic Shaping → **Pipe Rules** → **Add** → **Pipe Rule:**

→ **General**

Enter the following:

Name: FromInternet

Service: all-services

Address Filter

Source	Destination
--------	-------------

Interface: wan	lan
-----------------------	-----

Network: wannet	lannet
------------------------	--------

→ **Traffic Shaping**

Pipe Chains

Forward Chain: Select "std-in" from **Available** list and put it into **Selected** list.

Return Chain: Select "std-out" from **Available** list and put it into **Selected** list.

Precedence

Check **Use Fixed Precedence**

Select **Low** from the dropdown list and then click **OK**.



Example: Applying precedence on pipe limits

This example shows how to define specific precedences on pipes. We add one more rule on top of "ToInternet" and "FromInternet", which uses the two standard pipes created in the last example and enables Web browsing to the Internet to have higher priority "Medium" than all the other traffics having precedence "Low".

In order to prevent the service response from the Internet consuming all the bandwidth by its higher priority, a limit of 500 kbps is set into "Medium" precedence in pipe "std-in".

WebUI :

1. Additional pipe rule "HTTP" with fixed precedence "Medium":

Traffic Shaping → **Pipe Rules** → **Add** → **Pipe Rule:**

→ **General**

Enter the following:

Name: HTTP

Service: HTTP

Address Filter

Source	Destination
---------------	--------------------

Interface: lan	wan
-----------------------	-----

Network: lannet	wannet
------------------------	--------

→ **Traffic Shaping**

Pipe Chains

Forward Chain: Select "std-out" from **Available** list and put it into **Selected** list.

Return Chain: Select "std-in" from **Available** list and put it into **Selected** list.

Precedence

Check **Use Fixed Precedence**

Select **Medium** from the dropdown list and then click **OK**.

Right click the "HTTP" rule item and click **Move to Top**.

2. Revising pipe "std-in" to have a 500kbps limit on precedence "Medium"

Traffic Shaping → Pipes → std-in:

Pipe Limits

Precedences:

Add the following value into the edit box and then click **OK**.

Medium: 500



Example: Using grouping in a pipe

A pipe can be further divided into several groups with regard to particular network, IP, port, or interface; and the total bandwidth of the pipe can be fairly distributed onto each group by enabling Dynamic Bandwidth Balancing. The precedences applied to the pipe will also be used in all the groups. In this example, we revise the two standard pipes "std-in" and "std-out" to have grouping features based on Destination IP and Source IP respectively.

WebUI :

1. Editing pipe "std-in"

Traffic Shaping → Pipes → std-in:

Grouping

Grouping: Select **DestinationIP** from the dropdown list.

Check **Enable dynamic balancing of groups** and then click **OK**.

2. Editing pipe "std-out"

Traffic Shaping → Pipes → std-out:

Grouping

Grouping: Select **SourceIP** from the dropdown list.

Check **Enable dynamic balancing of groups** and then click **OK**.



Example: Using chains to create differentiated limits

More than one pipe can be connected into a pipe chain to make bandwidth limits more restrict. In the previous example—*Applying precedence on pipe limits*, a 500kbps limit on precedence "Medium" is defined on pipe "std-in". The "HTTP" rule says that HTTP response from the Internet can use up to 500kbps as higher priority traffic, and the traffic exceeding this limit will fall into priority "Low" specified by the standard rule "FromInternet". Such traffic will compete the remaining 500kbps with all the other traffics(The total limits defined for "std-in" is 1000kbps).

If we want to guarantee that other traffics always have at least 500kbps without competing with the exceeded HTTP traffic, we can add an additional pipe "http-in" that limits the total bandwidth consumption to 500kbps, and revise the pipe rule "HTTP" to have a pipe chain on the return direction. In this chain, "http-in" is put in front on "std-in". Traffic belongs to HTTP will need to pass the total limits in "http-in" first before it can go into "std-in". Exceeded HTTP traffic will be queued on "http-in".

WebUI :

1. Adding one more pipe "http-in" with total limits 500kbps

Enter the following and then click **OK**.

Traffic Shaping → **Pipes** → **Add** → **Pipe**:

General

Name: http-in

Pipe Limits

Total: 500

2. Revising pipe rule "HTTP" to create a return pipe chain

Traffic Shaping → **Pipe Rules** → **HTTP**

→ **Traffic Shaping**:

Pipe Chains

Return Chain:

Select "http-in" from **Available** list and put it into **Selected** list *on TOP of "std-in"*

and then click **OK**.

■  Note ■

- An appropriate order for pipes in a chain must be set carefully. ■

CHAPTER 24

Server Load Balancing (SLB)

24.1 Overview

Server Load Balancing (SLB) is a mechanism dealing with distributing the traffic load across multiple servers to scale beyond the capacity of one single server, and to tolerate a server failure. This technology is integrated in D-Link firewalls to enable high performance and throughput of the network.

24.1.1 The SLB Module

In the SLB module, a network appliance acts as a *Server load balancer*, connecting the network where the request traffic comes from with a cluster of servers called *Server farm*.

SLB logical view

Figure 24.1 illustrates a logical view of a SLB module. In this module, 3 servers construct a *server farm*, and a D-Link firewall acts as a *server load balancer*.

Server farm

A collection of computer servers usually maintained by an enterprise to accomplish the service needs far beyond the capability of a single machine.

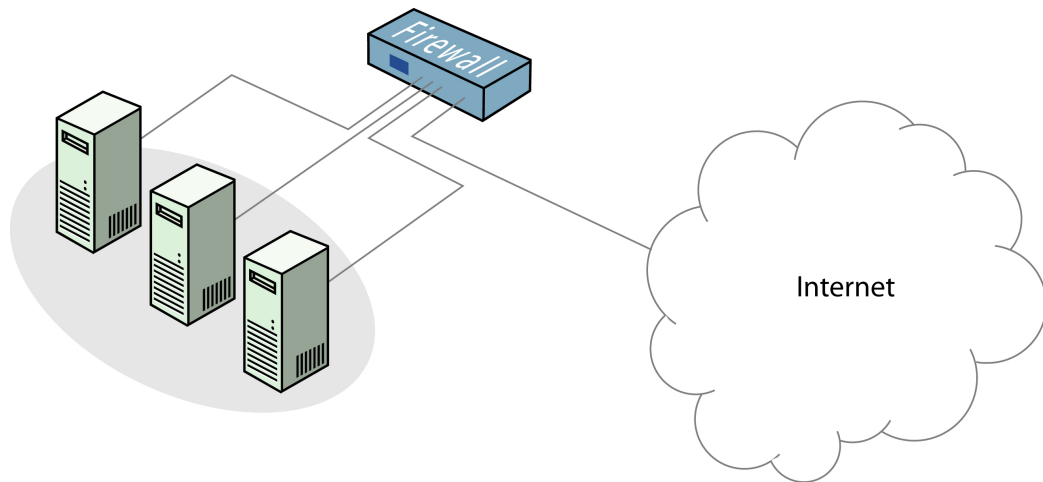


Figure 24.1: A SLB Logical View.

Server load balancer

An appliance to perform the functions of SLB, listening to the coming requests, deciding the traffic distribution mode and algorithm, rerouting the traffic to a certain sever within the server farm, and monitoring the availability of the servers.

D-Link firewalls are capable server load balancers, which can be configured to perform load distribution and monitoring functions.

24.1.2 SLB Features

The key features that SLB can provide are summarized as follows:

Load distribution

The *Load distribution* feature is responsible for distributing traffic to destination servers according to some predefined policy, i.e. distribution mode & algorithm. It determines *where* the traffic goes to and *how* is the traffic load shared among available servers.

Server Monitoring

Server Monitoring is used for performing various checks to evaluate the "health" of servers. It works at different layers of the OSI module, real-time

tracking the status of the servers, and noticing the load distribution to redirect traffic if there is any server failure.

24.1.3 Benefits

The SLB solution provides more advanced and flexible traffic management, and stronger processing power, compared to a single sever implementation. The significant advantages are:

Scalability

SLB dramatically improves the scalability of an application or server farm by distributing the load across multiple servers. The addition of new servers, and the removal or failure of existing servers, can occur at any time on demand, without experiencing downtime.

Optimized Capability

SLB helps to reduce the workload of each server, and hence, gives faster response to users' requests. Any single server in the server farm will not be overwhelmed by unusually heavy traffic that it is not capable to handle. The additional load can be taken over by other active servers automatically.

Availability

Load distribution and server monitoring cooperate to achieve automatic failover. With these two features, SLB is able to direct the traffic to alternative servers if a server fails.

Security

In SLB module, a public server address is present to the clients, representing the server farm. The real addresses of the servers are *hid* behind such public address and are never disclosed to the external network(covered by [24.2 SLB Implementation](#)). It can filter unwanted traffic based on both IP address and TCP or UDP port numbers, and helps to protect against multiple forms of denial-of-service(DoS) attacks.

Ease of Maintenance

Administration of server applications is easier. The sever farm is seen as a single virtual server by the Clients with one public address; no administration is required for real server changes, which are transparent to the external network.

24.2 SLB Implementation

To implement the SLB method, the administrator defines a *server farm* containing multiple real servers, and binds the server farm as a single virtual server to the D-Link firewall (load balancer), using a public IP address. In this environment, clients are configured to connect to the public address of the sever farm. When a client initiates a connection to the server farm, the firewall uses the *SAT rule* to translate the destination address. Which real servers will be chosen by the firewall as the most appropriate ones is determined by *predefined mode and algorithm*. Cooperating with the distribution task, the firewall monitors the "health" of the servers through some *layer 3/layer 4 connection checks*.

24.2.1 Distribution Modes

D-Link firewalls could be configured to work for SLB with the following modes:

1. *Per-state Distribution*:
The state of every distribution is recorded by the firewall. A complete session would be transferred to the same server to guarantee reliable data transmission.
2. *IP Address Stickiness*:
Sticky modes track the client connections to provide transmission integrity. In IP address stickiness mode, new connections from a client IP address are assigned to the same real server as were previous connections from that address.
3. *Network Stickiness*:
This mode works as the same as IP address stickiness, just apply to subnetwork addresses.

24.2.2 Distribution Algorithms

As advanced server load balancers, D-Link firewalls use configurable algorithms as selection criterion to control the traffic distribution. The firewall intelligently chooses the most appropriate servers and aims to maximize the total utilization of the server farm.

D-Link firewalls offer the following algorithms to accomplish the load distribution tasks:

1. **Round-Robin Algorithm** – treats all real servers as having equal capabilities, regardless of other facts, such as the number of existing connections or response time.
2. **Connection-Rate Algorithm** – redirects a connection to the server with the least number of new connections in a predefined time span. An array inside the firewall saves the number of new connections per second for each server. It updates every second to remove old connection counting values.

The *Round-Robin Algorithm* is suitable when the real servers within the server farm have equal processing powers, while using *Connection-Rate Algorithm* can optimize the response time.

Regardless which algorithm is chosen, if a server goes down, traffic will be sent to other servers. And when the sever comes back online, it can automatically be placed back into the server farm and start getting requests again.

24.2.3 Server Health Checks

Performing various checks to determine the "health" condition of servers is one of the most important benefits of the SLB. At different OSI layers, D-Link firewalls can carry out certain network-level checks.

When a server fails, the firewall removes it from the active server list, and will not route any packet to this server until it resumes back. An *ICMP Destination Host Unreachable message* will be sent by the firewall once the active server list is empty.

ICMP Ping

At OSI layer 3, the check involves a *Ping* to the real server's IP address to see whether the server is up and running.

TCP Connection

At OSI layer 4, the firewall attempts to connect to a configured port of the server where an application is running. For example, if the server is running web application (HTTP) on port 80, the firewall will try to establish a connection to bind to that port. It sends a TCP SYN request to port 80 on that server and waits for a TCP SYN/ACK in return; if failing, it marks the port 80 to be down on that server.

24.2.4 Packets Flow by SAT

In D-Link firewalls, *load-balancing enabled SAT* rule is used to translate packets exchanged between a client and real servers. When a new connection is being opened, the SAT rule is triggered; it translates the public server farm IP address to a real server address. Necessary modification to the packets is performed by the underlying system determined by *NAT* or *Allow* rule.

24.3 Scenario: Enabling SLB

The main configuration steps necessary for enabling SLB function in D-Link firewalls are outlined as follows:

- **Specifying a *Server Farm*** – Define a group of servers as a server farm by selecting the objects with correct IP.
- **Specifying the *load-balancing enabled SAT* rule** – Configure a SAT rule with filtering fields for the firewall to match the traffic flow and trigger the SLB .
- **Specifying *Distribution Mode & Algorithm*** – Provide the distribution policies the firewall should use.



Example: SLB Configuration

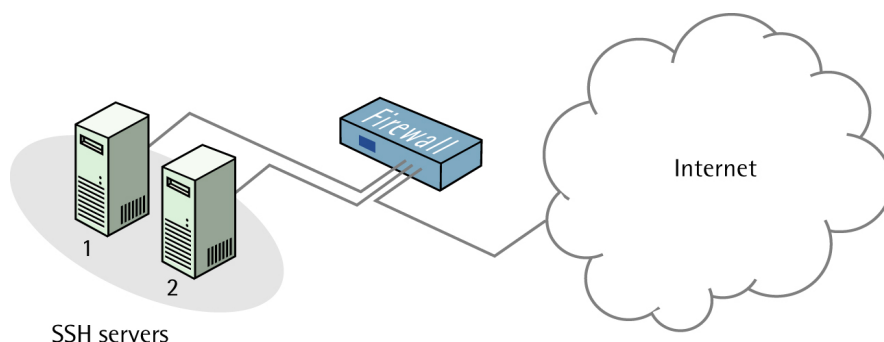


Figure 24.2: A SLB Scenario

This example describes how SLB can be used to load balance SSH connections to two SSH servers behind a D-Link Firewall connected to the Internet with IP address *ip_ext*, as shown in Figure 24.2. The two SSH servers have the private IP addresses 192.168.1.10 and 192.168.1.11.

WebUI :

1. Create Objects

First of all we need to create local objects to keep the IP address for each server.

Objects → **Address Book** → **Add** → **IP4 Host/Network**:

Name: SSH_Server1

IP Address: 192.168.1.10

Then click **OK**

Objects → **Address Book** → **Add** → **IP4 Host/Network**:

Name: SSH_Server2

IP Address: 192.168.1.11

Then click **OK**

2. Create SAT SLB Rule

Next step is to set up the SAT SLB rule.

Rules → **IP Rules** → **Add** → **IP Rule**:

Name: SSH_SLB

Action: SLB_SAT

Service: ssh

Source Interface: any

Source Network: all-nets

Destination Interface: core

Destination Network: ip_ext

SAT Server Load Balancing

Server Addresses: Select SSH_Server1 and SSH_Server2.

Then click **OK**

3. Create NAT Rule

Next step is to set up the NAT rule to permit traffic NAT:ed by the above rule.

Rules → **IP Rules** → **Add** → **IP Rule**:

Name: SSH_SLB_NAT

Action: NAT

Service: ssh

Source Interface: any

Source Network: all-nets

Destination Interface: core

Destination Network: ip_ext

Then click **OK**



- It is possible to configure settings for monitoring, distribution method and stickiness. But in this example the default values are used. ■

Part X

Misc. Features

Besides safety protection to the network, D-Link firewalls can act as intermediary agents for miscellaneous Internet services to ease the use of various protocols on behalf of the clients.

Topics in this part includes:

- [Miscellaneous Clients](#)
- [DHCP Server & Relay](#)

CHAPTER 25

Miscellaneous Clients

25.1 Overview

D-Link firewalls offer supports to miscellaneous network clients for *Dynamic DNS* and similar services. Currently, the services providers that are supported by the firewall include:

- Dyndns.org
- Dyns.cx
- Cjb.net
- Oray.net – Peanut Hull DynDNS
- Telia
- BigPond

25.2 Dynamic DNS

Dynamic Domain Name System (DynDNS), is a method of keeping a domain name linked to a changing IP address. When a user connects to the Internet through a means provided by the ISP, an unused IP address from a pool of IP addresses is assigned to the user's machine, and this address is used only for the duration of that specific connection. A dynamic DNS service provider uses a special program that runs on the user's machine,

contacting the DNS service each time the IP address provided by the ISP changes and subsequently updating the DNS database to reflect the change in IP address. This method allows the user's machine to have a domain name that always points to it, even though the IP address will change often. Other users do not have to know the changed IP address in order to connect to the machine.

In order to use this function as a DynDNS client, one must have an account with one of the supported service providers.

Dyndns.org

Dyndns.org is a free DynDNS service that allows registration under dozens of domains, e.g. "MYDNS.dyndns.org", "MYDNS.dnsalias.net", etc.

Dyns.cx

Dyns.cx is a free DynDNS service that allows registration under a number of domains: "dyns.cx", "dyns.net", "ma.cx", "metadns.cx", etc.

Cjb.net

Cjb.net provides free DynDNS service (and more) that allows registration under "cjb.net".

Oray.net

Oray.net – "Peanut Hull DynDNS" offers free DynDNS service under various domain names.

After successful register in one of the DynDNS service providers, a DynDNS client can configure the account information into the firewall to be able to automatically login to the service.

25.3 Automatic Client Login

Some Internet service providers require users to login via a URL each time before any service is delivered.

Currently, D-Link firewalls offers automatic client login to the following providers:

- **Telia** – A major telecommunication service company in the Nordic and Baltic region.

- **BigPond** – Used by Telstra, a broadband and multimedia service provider. Authenticates using the interface (which should be DHCP enabled) associated with the default route.

25.4 HTTP Poster

HTTP Poster is a function to enable *automatic* client login, or domain names and IP addresses update for DynDNS. When the firewall has parsed its configuration, the HTTP Poster posts all configured URLs in turn, and will wait a configurable delay time to re-post the URLs.



- Updating too often may cause the service provider to cancel the service. Thus, depending on the requirements by particular providers, the value of the delay should not be too small. ■

25.4.1 URL Format

The URL format used in the HTTP Poster varies depending on the specific service provider. Basically, a URL contains *Username/Password*, *provider's domain name*, and other parameters. For example, the URL format for DynDNS service provided by Dyndns.org is:

```
http://MYUID:MYPWD@members.dyndns.org/nic/update?hostname=MYDNS.dyndns.org
```


CHAPTER 26

DHCP Server & Relay

26.1 DHCP Server

The DHCP server implement the task to assign and manage IP addresses from specified address pools to DHCP clients. When a DHCP server receives a request from a DHCP client, it returns the configuration parameters (such as an IP address, a MAC address, a domain name, and a lease for the IP address) to the client in a unicast message. Because the DHCP server maintains configurations for several subnets, an administrator only needs to update a single, central server when configuration parameters change.

Compared to the static assignment where the client owns the address, dynamic addressing by the DHCP server leases the address to each client for a pre-defined period of time. During the life cycle of the lease, the client has permission to keep the assigned address and is guaranteed to have no address collision with other clients. Before the expiration of the lease, the client needs to *renew* the lease from the server, so it can keep using its IP address. The client may also decide at any time that it no longer wishes to use the IP address it was assigned, and may terminate the lease by *releasing* the IP address. The lease time can be configured in the DHCP server by the administrator.



Example: Configuring the firewall as a DHCP server

This example describes how to configure a DHCP server on the internal interface (LAN)(Refer to 9.1.2, Ethernet Interfaces in D-Link Firewalls).

WebUI :

- **Configure DHCP Server**

System → **DHCP Settings** → **DHCP Server** → **Add** → **DHCP Server:**

Enter the following:

Name: dhcpserver_lan

Interface Filter: LAN (The interface(s) to listen for DHCP requests on)

IP Address Pool: 192.168.1.10-192.168.1.20 (The pool of IP addresses to hand out)

Netmask: 255.255.255.0 (Specify the netmask to hand out)

Options

Default GW: Specify the default gateway to hand out to DHCP clients. In this case (None).

Domain: Specify the domain to hand out. This can be left empty.

Lease Time: Configure the time a lease should be valid.

DNS: Configure the DNS server information to hand out to DHCP clients. This can be left to (None).

NBNS/WINS: Configure the NBNS/WINS server information to hand out to DHCP clients. This can be left to (None).

Next Server: Specifies the IP address of next server in the boot process, this is usually a TFTP server. This can be left to (None).

Custom Options

Here you can add custom options to the DHCP lease. It is possible to specify the code, type and parameter.

When finished, click **OK**

26.2 DHCP Relay

In DHCP implementation, the clients send requests to locate the DHCP server(s) by broadcast messages. However, broadcasts are normally only propagated on the local network. This means that the DHCP server and client would always need to be in the same physical network area to be able to communicate. In such a case, for a large Internet environment, it requires a different server on every network, and the benefit of having one centralized server configuration is greatly reduced. This problem is solved by the use of DHCP relayer.

A DHCP relayer takes the place of the DHCP server in the local network to act as the intermedium between the client and the remote DHCP server. It intercepts requests from clients and relays them to the server. The server then responds back to the relay, which forwards the response to the client. The DHCP relayers follow the BOOTP relay agent functionality and retain the BOOTP message format and communication protocol, and hence, they are often called BOOTP relay agents.



Example: Configuring the firewall as a DHCP relay

Configuration in this example allows clients on the VLAN interfaces to obtain IP addresses from a DHCP server.

Before the following steps are taken, it is assumed that the firewall is configured with VLAN interfaces that are going to use DHCP relaying, and the IP address of the DHCP server has been defined in the address book named as "ip-dhcp".

For information about VLAN configuration, please refer to [9.2.3, VLAN Implementation](#). In this case, two VLAN interfaces named as "vlan1" and "vlan2" are used.

The firewall will also install a route for the client when it has finalized the DHCP process and obtained an IP.

WebUI :

1. Interface group:

– adding the VLAN interfaces "vlan1" and "vlan2" that should relaying to an interface group named as "ipgrp-dhcp".

Interface → **Interface Groups** → **Add** → **Interface Group:**

Name: ipgrp-dhcp

Interfaces: select "vlan1" and "vlan2" from the **Available** list and put them into the **Selected** list.

Then click **OK**.

2. DHCP relay:

– adding a DHCP relay named as "vlan-to-dhcpserver"

System → **DHCP Settings** → **DHCP Relays** → **Add** → **DHCP Relay:**

→ **General:**

General

Name: vlan-to-dhcpserver

Action: Relay

Source Interface: ipgrp-dhcp

DHCP Server to relay to: ip-dhcp

→ **Add Route:**

Check **Add dynamic routes for this relayed DHCP lease.**

Then click **OK**.

Part XI

Transparent Mode

CHAPTER 27

Transparent Mode

The *Transparent Mode* feature provided by D-Link firewalls aims at simplifying the deployment of firewall appliances into the existing network topology, to strengthen security. It helps to ease the administration work in a way that there is no need to reconfigure all the settings for the nodes within the current network, when a firewall is introduced into the communication flow.

In this chapter we give you an overview of the transparent mode feature and introduce how transparent mode is implemented in D-Link firewalls in detail. Configuration examples of simple network layouts and more complicated real-life scenarios can be found at the end of this chapter.

27.1 Overview

Transparency refers to the visibility of the firewall to hosts on both side of a firewall. A firewall is considered transparent to users if they do not notice the firewall in the packet flow. When adding a transparent firewall into a preexisting network structure, we achieve the following advantages for network administrators:

- No reconfiguration required – clients can keep the same network configuration after the firewall has been installed.
- No obstacle added – the deployment of the firewall should be *invisible* to the internal users, as they can still obtain the allowed services.

- Enhanced security – the firewall should be capable of screening the in/out traffic by the defined security rules.

D-Link firewalls can work in two modes: *Routing Mode* & *Transparent Mode*. In normal *Routing Mode*, the firewall acts as a Layer 3 router. If the firewall is placed into a network for the first time, or if there is any topological change within the nodes, the routing configuration must be thoroughly examined to ensure that the routing table of the firewall system is consistent with the current network layout. Reconfiguration of IP settings is also required for preexisting routers and protected servers. This mode works well when we want to have complete control over routing, and be aware of the specific location of important devices, to have the highest possible security. For instance, we expect that a server located at a protected area only receives necessary traffic.

While in the *Transparent Mode*, the firewall acts more like a switch. It screens IP packets traversing the firewall and forwards them transparently on the right interface without modifying any of the source or destination information. All transparent interfaces are considered to be in the same network, so if one client moves to another interface it can still obtain the same services as before without routing reconfiguration.

In transparent mode, the firewall allows ARP transactions over the firewall, and learns from ARP traffic the relation between the IP address and the physical address of the source and destination. There are mechanisms helping the firewall to remember the address information, in order to relay IP packets to the desired receiver. During the transaction, none of the endpoints will be aware of the firewall working in between.

27.2 Transparent Mode Implementation in D-Link Firewalls

As explained above, D-Link firewall allows ARP transactions when it is set to be transparent mode and in that sense it works almost as a Layer 2 switch in the network. The firewall uses the ARP traffic as one source of information when building its switch route table. To start with the transparent mode, the following setup needs to be done in the firewall:

- Group the interfaces – specify a group of interfaces that are going to use transparent mode.

- Create a *Switch Route* – as interface, select the interface group created earlier. As network, specify the address range that should be transparent between the interfaces. When the whole firewall is working in Transparent Mode this is normally 0.0.0.0/0.

When initiating communication, a host will locate the other host's physical address by broadcasting an ARP request. When the firewall intercepts an ARP request, it sets up a *ARP Transaction State* inside the firewall and broadcasts the ARP request to all the other switch-route interfaces except the interface the ARP request was received on. If the firewall receives an ARP reply from the destination within a three second timeout, it will relay the reply back to the sender of the request, using information stored in the ARP Transaction State.

During the ARP transaction, the firewall learns the source address information of both ends from the request and reply. Inside the D-Link firewall, two tables are maintained that are used to store such information, called *Content -Addressable Memory(CAM) Table* and *Layer 3 Cache* respectively.

The CAM table contains information of the MAC addresses available on a given physical interface of the firewall, while the Layer 3 cache stores mappings between IP address, MAC address and interface.

As the Layer 3 Cache is only used for IP traffic, Layer 3 Cache entries are stored as single host entries in the routing table.

For each IP packet that will traverse the firewall, a route lookup for the destination will be done. If the route of the packet matches a switch route or a Layer 3 Cache entry in the routing table, the firewall knows that it should handle this packet in a transparent manner. If a destination interface and MAC address is available in the route, the firewall has the necessary information to forward the packet to the destination. If the route was a switchroute, no specific information about the destination is available and the firewall will have to discover where the destination is located in the network. Discovery is done by sending out ARP requests, acting as the initiating sender of the original IP packet, for the destination on the interfaces specified in the switchroute. If a ARP reply is received, the firewall will update the CAM table and Layer 3 Cache and forward the packet to the destination.

If the CAM table or the Layer 3 Cache is full, the tables are partially flushed automatically. Using the discovery mechanism, the firewall will

rediscover destinations that may have been flushed.

27.3 Scenarios: Enabling Transparent Mode



Example: Transparent Mode Scenario 1

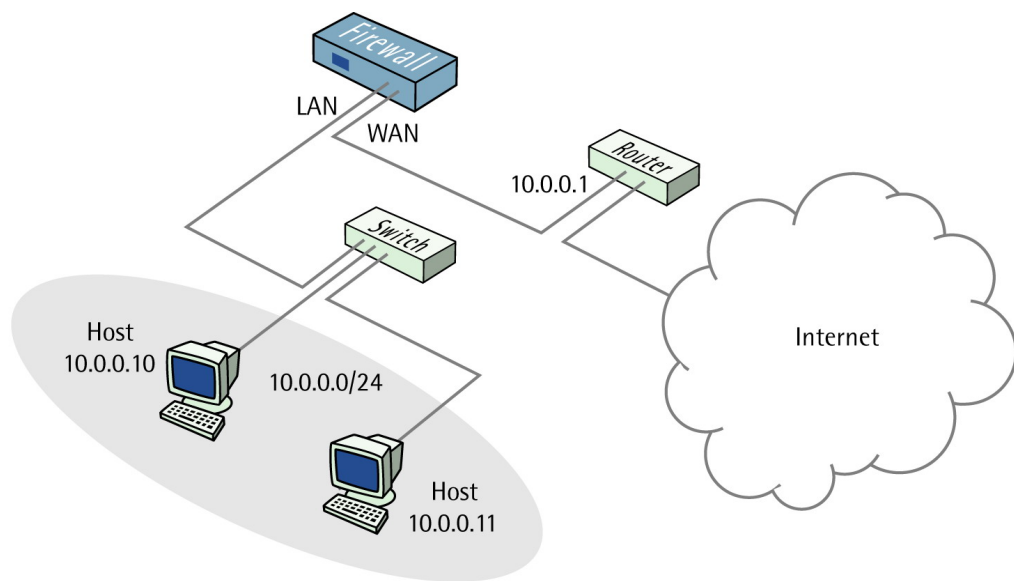


Figure 27.1: Transparent Mode Scenario 1.

Scenario 1 shows how a firewall in Transparent Mode can be placed in an existing network between an Internet access router and the internal network, without the need to reconfigure clients in the internal network.

In this scenario a router is used to share an Internet connection with a single public IP address. The internal NAT:ed network behind the firewall is in the *10.0.0.0/24* address space. Clients on the internal network should be allowed to access the Internet via the HTTP protocol.

The *WAN* and *LAN* interfaces of the firewall will have to be configured to operate in Transparent Mode. It is preferred to configure IP addresses on the *WAN* and *LAN* interfaces, as this can improve performance during automatic discovering of hosts.

All traffic passing through the firewall will have to pass through the IP rule set. To allow HTTP traffic, a new IP rule has to be configured. (Refer to [14.3 Scenario](#).)

WebUI :

1. Interfaces

Interfaces → Ethernet → Edit (WAN):

Enter the following:

IP Address: 10.0.0.2

Network: 10.0.0.0/24

Default Gateway: 10.0.0.1

Transparent Mode: Enable

Then click **OK**

Interfaces → Ethernet → Edit (LAN):

Enter the following:

IP Address: 10.0.0.2

Network: 10.0.0.0/24

Transparent Mode: Enable

Then click **OK**

2. Rules

Rules → IP Rules → Add → IP Rule:

Enter the following:

Name: HTTPAllow

Action: Allow

Service: http

Source Interface: LAN

Destination Interface: any

Source Network: 10.0.0.0/24

Destination Network: 0.0.0.0/0 (all-nets)

Then click **OK**


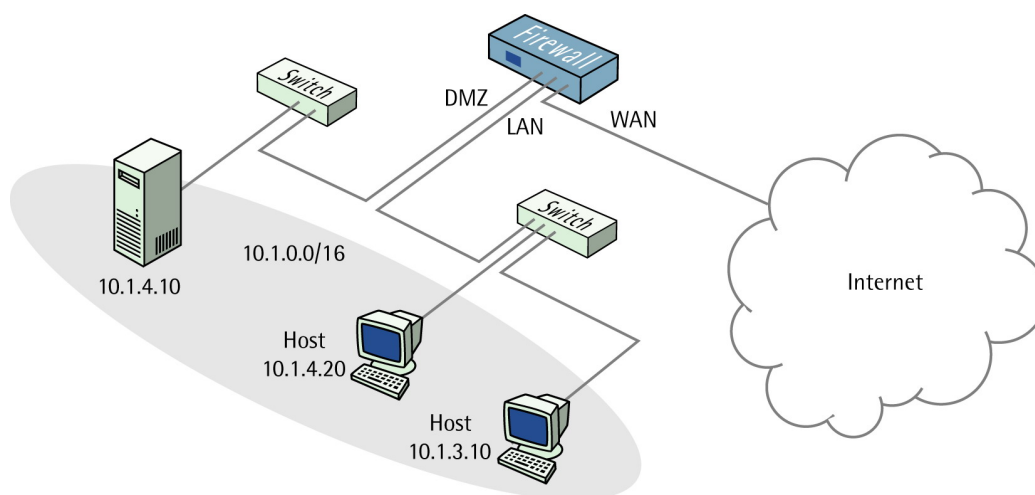
 Example: Transparent Mode Scenario 2

Figure 27.2: Transparent Mode Scenario 2.

Scenario 2 shows how a firewall in Transparent Mode can be used to separate server resources from the internal network by attaching them to a separate firewall interface without the need of different address ranges.

Servers containing resources that is accessible from the outside could be a security risk if they are placed directly on the internal network. Because of this, such servers are often connected to a separate interface on the firewall, like *DMZ*.

In this scenario all hosts connected to *LAN* and *DMZ* shares the the *10.0.0.0/24* address space. As this is configured using Transparent Mode any IP address can be used for the servers, and there is no need for the hosts on the internal network to know if a resource is on the same network or placed on *DMZ*. This makes the firewall transparent in the communication between *DMZ* and *LAN* even though the traffic can be restricted using the *firewall IP ruleset*.

Here we allow the hosts on the internal network to communicate with an HTTP server on *DMZ*. Furthermore, we allow the HTTP server on *DMZ* to be reached from the internet. Additional rules could be added to allow

other traffic.

This scenario shows how to configure a Switch Route over the *LAN* and *DMZ* interfaces for the *10.0.0.0/24* address space.

It is assumed that the WAN interface is configured correctly already.

WebUI :

1. Interfaces

Interfaces → Ethernet → Edit (LAN):

Enter the following:

IP Address: 10.0.0.1

Network: 10.0.0.0/24

Transparent Mode: Disable

Add route for interface network: Disable

Then click **OK**

Interfaces → Ethernet → Edit (DMZ):

Enter the following:

IP Address: 10.0.0.2

Network: 10.0.0.0/24

Transparent Mode: Disable

Add route for interface network: Disable

Then click **OK**

2. Interface Groups

Interfaces → Interface Groups → Add → Interface Group:

Enter the following:

Name: TransparentGroup

Security/Transport Equivalent: Disable

Interfaces: Select LAN and DMZ

Then click **OK**

3. Routing

Routing → Main Routing Table → Add → Switch Route:

Enter the following:

Switched Interfaces: TransparentGroup

Network: 10.0.0.0/24

Metric: 0

Then click **OK**

4. Rules

Rules → IP Rules → Add → IP Rule:

Enter the following:

Name: HTTP-LAN-to-DMZ

Action: Allow

Service: http

Source Interface: LAN

Destination Interface: DMZ

Source Network: 10.0.0.0/24

Destination Network: 10.1.4.10

Then click **OK**

Rules → IP Rules → Add → IP Rule:

Enter the following:

Name: HTTP-WAN-to-DMZ

Action: SAT

Service: http

Source Interface: WAN

Destination Interface: DMZ

Source Network: 10.0.0.0/24

Destination Network: wan-ip

Translate: Select Destination IP

New IP Address: 10.1.4.10

Then click **OK**

Rules → IP Rules → Add → IP Rule:

Enter the following:

Name: HTTP-WAN-to-DMZ

Action: Allow

Service: http

Source Interface: WAN

Destination Interface: DMZ

Source Network: 10.0.0.0/24

Destination Network: wan-ip

Then click **OK**

Part XII

Zone Defense

CHAPTER 28

Zone Defense

28.1 Overview

Zone Defense is a feature in D-Link firewalls, which lets the firewall control locally attached switches. This can be used as a countermeasure to stop a worm-infected computer in the local network from infecting other computers.

By setting up *threshold rules* on the firewall, hosts or networks that are exceeding the defined threshold can be dynamically blocked out. The thresholds are based on the number of new connections made per second by either a single host or all hosts within a specified CIDR network range (an IP address range specified by a combination of an IP address and its associated network mask). When the firewall notices that a host or a network has reached the specified limit, it uploads ACL (Access Control List) rules to the switches, which in turn blocks all traffic for that host or network. Blocked hosts and networks remain blocked until the system administrator *manually* unblocks them using the firewall's Web or command line interface.

28.2 Zone Defense Switches

Switch information regarding every switch that is to be controlled by the firewall has to be manually specified in the firewall configuration. The information needed in order to control a switch includes:

- The IP address of the management interface of the switch
- The switch model type
- The SNMP community string (write access)

Currently, Zone Defense feature supports the following switches:

- D-Link DES 3226S (minimum firmware: R4.02-B14)
- D-Link DES 3250TG (minimum firmware: R3.00-B09)
- D-Link DES 3326S (minimum firmware: R4.01-B39)
- D-Link DES 3350SR (minimum firmware: R1.02.035)
- D-Link DES 3526 (minimum firmware: R3.01-B23)
- D-Link DES 3550 (minimum firmware: R3.01-B23)
- D-Link DGS 3324SR (minimum firmware: R4.10-B15)



- Make sure that the switches have the minimum required firmware versions before activating Zone Defense. ■

28.2.1 SNMP

Simple Network Management Protocol (SNMP) is an application layer protocol for complex network management. SNMP allows *managers* and *managed devices* in a network to communicate by sending messages, for the purpose of accessing different parts of the network.

Manager

A typical manager, such as a D-Link firewall, executes SNMP protocol to monitor and control network devices in the managed environment. The manager can query stored statistics from the controlled devices by using the ***SNMP community string***, which is like a user id or password to allow access to the device's database. If the community string type is "*write*", the manager will be allowed to modify properties in the device.

Managed devices

The managed devices are SNMP compliant, such as D-Link switches. They store management data in their databases, known as *Management Information Base (MIB)*, and provide the information to the manager upon queries.

28.3 Threshold Rules

As explained previously, a threshold rule will trigger Zone Defense to block out a specific host or a network if the connection rate limit specified in the rule is exceeded. Similar to the IP rules, a threshold rule also contains several fields, specifying which type of traffic that should match the rule.

In total, a threshold rule is defined by:

- Source interface and source network.
- Destination interface and destination network.
- Service.
- Type of threshold: Host and/or network based.

Traffic that matches the criterion above and causes the host/network threshold to be exceeded will trigger Zone Defense function, which will prevent the host/networks from accessing the switch(es). All blocks in response to threshold violations will be prohibited based on IP address of the host or network on the switch(es). When a network-based threshold has been exceeded, the source network will be blocked out instead of just the offending host.

28.4 Manual Blocking & Exclude Lists

As a complement to the threshold rules, it is also possible to manually define hosts and networks that are to be statically blocked or excluded. Manually blocked hosts and networks can be blocked by default or based on a schedule. It is also possible to specify which protocols and protocol port numbers that are to be blocked.

Exclude lists can be created and used in order to exclude hosts from being blocked when a threshold rule limit is reached. Good practice includes

adding the firewall's interface IP or MAC address connecting towards the Zone Defense switch to the Exclude list. This prevents the firewall from being accidentally blocked out.

28.5 Limitations

Depending on the switch model, various limitations are in effect. The first one is the latency between the triggering of a block rule to the moment of the switch(es) actually blocking out the traffic matched by the rule. All switch models require at least some time to enforce the rules after they have been provided by the firewall. Some models can activate the rules within a second while others require up to a minute or even beyond.

Another limitation is the maximum number of rules supported by the switch. Some switches support only 50 rules while others support up to 800 (usually, in order to block a host or network, *one rule per switch port* is needed). When this limit has been reached no more hosts or networks will be blocked out.

Zone Defense uses the ACL rule set on the switch and will initially purge all entries on the switch. All pre-configured ACLs will be *lost*.

28.6 Scenario: Setting Up Zone Defense

The following simple example illustrates the steps needed to set up Zone Defense function in D-Link firewalls. We assume that all the interfaces on the firewall have already been properly configured.



Example: Configuring Zone Defense

In this simplified scenario, a HTTP threshold of 10 connections/second is applied. If the connections exceed this limitation, the firewall will block the specific host (in network range 192.168.2.0/24 for example) from accessing the switch completely.

A D-Link switch model DES-3226S is used in this case, with a management interface address 192.168.1.250 connecting to the firewall's interface address 192.168.1.1. This firewall interface is added into the exclude list to prevent

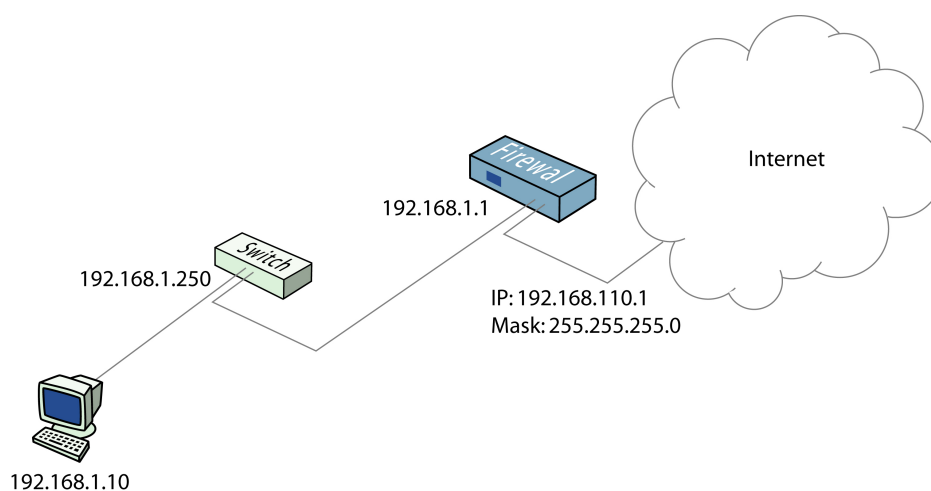


Figure 28.1: A Zone Defense Scenario.

the firewall from being accidentally locked out from accessing the switch. The simplified network layout for this scenario is shown in Figure 28.1.

WebUI :

1. Switch

– adding a new switch into Zone Defense section.

Zone Defense → Switches → Switch:

General

Name: switch1

Switch model: DES-3226S

IP Address: 192.168.1.250

(or use the object name if it has been defined in the address objects)

SNMP Community:

Enter in the edit box the *write community string* configured on the switch.

Press **Check Switch** button to verify that the firewall can communicate with the switch and the correct community string is entered.

Then click **OK**.

2. Exclude list

– Adding the firewall’s management interface into the exclude list.

Zone Defense → Exclude:

General**Addresses:**

Choose the object name of the firewall’s interface address 192.168.1.1 from **Available** list and put it into **Selected** list.

Then click **OK**.

3. Threshold rule

– configuring a HTTP threshold of 10 connections/second.

Zone Defense → Threshold → Add → Threshold:

→ **General:**

General:

Name: HTTP-Threshold

Service: HTTP

Address Filter**Source****Destination**

Interface: (the firewall’s management interface) any

Network: 192.168.2.0/24(or the object name) all-nets

→ **Action:**

Action: ZoneDefense

Host-based Threshold: 10

Then click **OK**.

Part XIII

High Availability

CHAPTER 29

High Availability

29.1 High Availability Basics

This section includes the following topics:

- What High Availability will do for you
- What High Availability will NOT do for you
- Example High Availability setup

D-Link High Availability works by adding a back-up firewall to your existing firewall. The back-up firewall has the same configuration as the primary firewall. It will stay inactive, monitoring the primary firewall, until it deems that the primary firewall is no longer functioning, at which point it will go active and assume the active role in the cluster. When the other firewall comes back up, it will assume a passive role, monitoring the now active firewall.

Throughout this chapter, the phrases "master firewall" and "primary firewall" are used interchangeably, as are the phrases "slave firewall" and "back-up firewall".

29.1.1 What High Availability will do for you

D-Link High Availability will provide a redundant, state-synchronized firewalling solution. This means that the state of the active firewall, i.e.

connection table and other vital information, is continuously copied to the inactive firewall. When the cluster fails over to the inactive firewall, it knows which connections are active, and communication may continue to flow uninterrupted.

The failover time is typically about one second; well in the scope for the normal TCP retransmit timeout, which is normally over one minute. Clients connecting through the firewall will merely experience the failover procedure as a slight burst of packet loss, and, as TCP always does in such situations, retransmit the lost packets within a second or two, and go on communicating.

29.1.2 What High Availability will NOT do for you

Adding redundancy to your firewall setup will eliminate one of the single points of failure in your communication path. However, it is not a panacea for all possible communication failures.

Typically, your firewall is far from the only single point of failure. Redundancy for your routers, switches, and your Internet connection are also issues that need to be addressed.

D-Link High Availability clusters will not create a load-sharing cluster. One firewall will be active, and the other will be inactive.

Multiple back-up firewalls cannot be used in a cluster. Only two firewalls, a "master" and a "slave", is supported.

As is the case with all other firewalls supporting stateful failover, the D-Link High Availability will only work between two D-Link Firewalls. As the internal workings of different firewalls, and, indeed, different major versions of the same firewall, can be radically different, there is no way of communicating "state" to something which has a completely different comprehension of what "state" means.

Broken interfaces will not be detected by the current implementation of the High Availability, unless they are broken to the point where the firewall cannot continue to run. This means that failover will not occur if the active firewall can communicate being alive to the inactive firewall through any of its interfaces, even though one or more interfaces may be inoperative.

29.1.3 Example High Availability setup

All the interfaces of the primary firewall need to be present on the back-up firewall, and connected to the same networks. As previously mentioned, failover is not done unnecessarily, so either firewall may maintain the active role of the cluster for an extended period of time. Hence, connecting some equipment to only the "master" or only the "slave" firewall is bound to produce unwanted results.

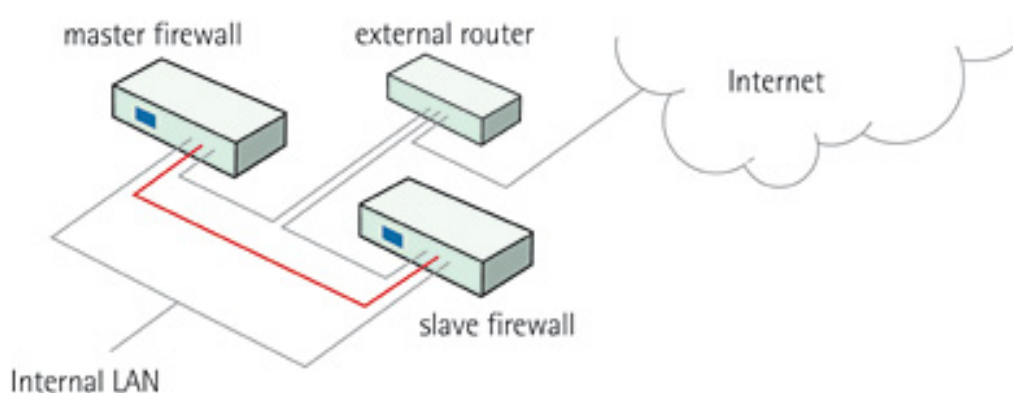


Figure 29.1: Example HA Setup.

As you can see in figure 29.1, both firewalls are connected to the internal as well as the external network. If there are more networks, for instance one or more demilitarized zones, or internal network segments, both firewalls will also have to be connected to such networks; just connecting the "master" to a network will most likely lead to loss of connectivity for extended periods of time.

29.2 How Rapid Failover is Accomplished

This section includes the following topics:

- The shared IP address and the failover mechanism
- Cluster heartbeats
- The synchronization interface

This section will detail the outward-visible characteristics of the failover mechanism, and how the two firewalls work together to create a high availability cluster with very low failover times.

For each cluster interface, there are three IP addresses:

- Two "real" IP addresses; one for each firewall. These addresses are used to communicate with the firewalls themselves, i.e. for remote control and monitoring. They should not be associated in any way with traffic flowing through the cluster; if either firewall is inoperative, the associated IP address will simply be unreachable.
- One "virtual" IP address; shared between the firewalls. This is the IP address to use when configuring default gateways and other routing related matters. It is also the address used by dynamic address translation, unless the configuration explicitly specifies another address.

There is not much to say about the real IP addresses; they will act just like firewall interfaces normally do. You can ping them or remote control the firewalls through them if your configuration allows it. ARP queries for the respective addresses are answered by the firewall that owns the IP address, using the normal hardware address, just like normal IP units do.

29.2.1 The shared IP address and the failover mechanism

Both firewalls in the cluster know about the shared IP address. ARP queries for the shared IP address, or any other IP address published via the ARP configuration section or through Proxy ARP, will be answered by the active firewall.

The hardware address of the shared IP address, and other published addresses for that matter, is not related to the hardware addresses of the firewall interfaces. Rather, it is constructed from the cluster ID, on the following form: 10-00-00-C1-4A-nn, where nn is the Cluster ID configured in the Settings section.

As the shared IP address always has the same hardware address, there will be no latency time in updating ARP caches of units attached to the same LAN as the cluster when failover occurs.

When a firewall discovers that its peer is no longer operational, it will broadcast a number of ARP queries for itself, using the shared hardware address as sender address, on all interfaces. This causes switches and bridges to re-learn where to send packets destined for the shared hardware address in a matter of milliseconds.

Hence, the only real delay in the failover mechanism is detecting that a firewall is no longer operational.

The activation messages (ARP queries) described above are also broadcast periodically to ensure that switches won't forget where to send packets destined for the shared hardware address.

29.2.2 Cluster heartbeats

A firewall detects that its peer is no longer operational when it can no longer hear "cluster heartbeats" from its peer.

Currently, a firewall will send five cluster heartbeats per second.

When a firewall has "missed" three heartbeats, i.e. after 0.6 seconds, it will be declared inoperative.

So, why not make it even faster? Maybe send a hundred heartbeats per second and declare a firewall inoperative after missing only two of them? This would after all result in a 0.02-second failover time.

The problem with detection times less than a tenth of a second is that such delays may occur during normal operation. Just opening a file, on either firewall, could result in delays long enough to cause the inactive firewall to go active, even though the other firewall is still active; a clearly undesirable situation.

Cluster heartbeats have the following characteristics:

- The source IP is the interface address of the sending firewall
- The destination IP is the shared IP address
- The IP TTL is always 255. If a firewall receives a cluster heartbeat with any other TTL, it is assumed that the packet has traversed a router, and hence cannot be trusted at all.

- It is an UDP packet, sent from port 999, to port 999.
- The destination MAC address is the ethernet multicast address corresponding to the shared hardware address, i.e. 11-00-00-C1-4A-nn. Link-level multicasts were chosen over normal unicast packets for security reasons: using unicast packets would have meant that a local attacker could fool switches to route the heartbeats somewhere else, causing the peer firewall to never hear the heartbeats.

29.2.3 The synchronization interface

Both firewalls are connected to each other by a separate synchronization connection; normal network cards are used, although they are dedicated solely for this purpose.

The active firewall continuously sends state update messages to its peer, informing it of connections that are opened, connections that are closed, state and life time changes in connections, etc.

When the active firewall ceases to function, for whatever reason and for even a short time, the cluster heartbeat mechanism described above will cause the inactive firewall to go active. Since it already knows about all open connections, communication can continue to flow uninterrupted.

29.3 Setting up a High Availability Cluster

This section includes the following topics:

- Planning the High Availability cluster
- Creating a High Availability cluster

This section describes the process of installing a High Availability cluster. For a successful installation, it is highly recommended that the previous sections, High Availability Basics and How rapid failover is accomplished, have been read.

A cluster can be created by either installing a pair of new firewalls, or by converting already installed firewalls to cluster members.

The firewall with the highest version number of its configuration will always make sure that the configuration is transferred to the other cluster member.

The topics below describe the operations required to setup a complete High Availability cluster.

29.3.1 Planning the High Availability cluster

As an example throughout this guide, two D-Link Firewalls are used as cluster members. To simplify this guide, only two of the interfaces on each cluster member are used for network traffic. The following setup is used:

- The LAN interfaces on the cluster members are both connected to the same switch. This switch resides on an internal network with IP addresses from the 192.168.10.0/24 network.
- The WAN interfaces on the cluster members are both connected to a second switch. This switch resides on an external network with IP addresses from the 10.4.10.0/24 network.
- The IP addresses for the interfaces are designated as indicated by this table:

Interface	Shared IP address	Master IP address	Slave IP address
LAN	192.168.10.1	192.168.10.2	192.168.10.3
WAN	10.4.10.1	10.4.10.2	10.4.10.3

- The DMZ interfaces on the cluster members are used for state synchronization, and therefore connected to each other using a crossover Ethernet cable.

29.3.2 Creating a High Availability cluster



Example: Configuring the Firewall as a Cluster Member

Each firewall in the cluster will have to be configured to act as either a HA master or slave. This includes configuration of private (master and slave) and shared IP addresses on interfaces, as well as selecting a cluster ID and synchronization interface.

WebUI :

1. HA Configuration

System → **High Availability**:

Enable High Availability: Enable

Cluster ID: 0 (Select a suitable cluster ID)

Sync Interface: DMZ (In this example we use the DMZ interface as sync interface)

Node Type: Master or Slave

Then click **OK**

2. HA Address Pair Configuration

We need to create HA Address Pair objects to store the master and slave IP addresses.

Objects → **Address Book** → **Add** → **HA IP4 Address Pair**:

Name: lan-priv-ip (Private IP addresses for LAN interface)

Master IP Address: 192.168.10.2

Slave IP Address: 192.168.10.3

Then click **OK**

Objects → **Address Book** → **Add** → **HA IP4 Address Pair**:

Name: wan-priv-ip (Private IP addresses for WAN interface)

Master IP Address: 10.4.10.2

Slave IP Address: 10.4.10.3

Then click **OK**

3. Interface Configuration

Interfaces → **Ethernet** → **Edit (LAN)**:

IP Address: 192.168.10.1

Advanced/High Availability

Private IP Address: lan-priv-ip Then click **OK**

Interfaces → **Ethernet** → **Edit (WAN)**:

IP Address: 10.4.10.1

Advanced/High Availability

Private IP Address: wan-priv-ip

Then click **OK**

When the configuration is saved and activated, the firewall will act as a HA cluster member.



■ All Ethernet and VLAN interfaces will have to be assigned a private IP address when the firewall is configured to be a HA member. However, in this example we only showed how to configure the LAN and WAN interfaces. Note that it is possible to use the same HA IP4 Address Pair object on multiple interfaces.

When a modification of the configuration on either of the firewalls has been saved and activated, the configuration will automatically be transferred to the other cluster member. It doesn't matter if the configuration was changed on the master or slave firewall, as the cluster member with the highest configuration version number will always try to transfer the configuration to the other cluster member. ■

29.4 Things to Keep in Mind

This section includes the following topics:

- Statistics and Logging Issues
- Configuration Issues

Even though your high availability cluster will behave like a single firewall from most aspects, there are a few things to keep in mind when managing and configuring it.

29.4.1 Statistics and Logging Issues

SNMP Statistics

SNMP statistics are not shared. SNMP managers have no failover capabilities. Thus, you will need to poll both firewalls in the cluster.

Logs come from two firewalls

Log data will be coming from two firewalls. This means that you will have to configure your log receiver to receive logs from both firewalls. It also means that all your log queries will likely have to include both firewalls as sources, which will give you all the log data in one result view. Normally, the inactive firewall won't be sending log entries about live traffic, so the output will likely look much the way it did with only one firewall.

29.4.2 Configuration Issues

When configuring High Availability clusters, there are a number of things to keep in mind in order to avoid unnecessary pitfalls.

Changing the cluster ID

By changing the cluster ID, you actually doing two things:

- Changing the hardware address of the shared IPs. This will cause problems for all units attached to the local LAN, as they will keep the old hardware address in their ARP caches until it times out. Such units will have to have their ARP caches flushed.
- You will also break the connection between the firewalls in the cluster for as long as they are using different configurations. This will cause both firewalls to go active at the same time.

In short, changing the cluster ID unnecessarily is not a good idea.

After the configuration has been uploaded to both firewalls, the ARP caches of vital units will have to be flushed in order to restore communication.

Never use the unique IPs for live traffic

The unique (private) IP addresses of the firewalls cannot safely be used for anything but managing the firewalls.

Using them for anything else: gatewaying, using them as source IPs in dynamically NATed connections or publishing services on them, will inevitably cause problems, as unique IPs will disappear when the firewall it belongs to does.

Part XIV

Appendix

INDEX

ABR, 75
ACL, 293
ActiveX, 156
AES, 196
AH, 214
ALG, 47
ARP, 27
ARP, 66
AS, 70
ASBRs, 75

Backbone area, 74
BDR, 75
Blowfish, 196
BOOTP, 60
Brute force attack, 196

CA, 49, 199
CAST, 196
Certificate, 49, 199, 221
CHAP, 62, 134, 135, 229
CRL, 50, 200, 222
Cryptography, 195

DES, 196
DH group, 218
DHCP, 60, 275, 277
Dictionary attack, 229
Diff-Serv, 247, 250
Digital signature, 198

DMZ, 14, 119, 150, 200, 204, 207,
210
DNS, 101
DoS, 47, 123, 263
DR, 75
DSA, 197
DST, 97
DynDNS, 271

ESP, 214
Ethernet, 53
Ethernet address, 41

Firewall, 9

GRE, 27, 45, 228

H.225, 160
H.245, 160
H.323, 158
High availability, 54
Hop, 69
HTTP, 155
HTTPPoster, 273
HTTPS, 46, 136, 243

IANA, 45
ICMP, 43
ID Lists, 221
IDlist, 51
IDS, 26

- IKE, 213
- IKE XAuth, 222
- IP address, 39
- IP spoofing, 123
- IPsec, 27, 213

- L2TP, 27, 228
- LAN, 53, 56
- LCP, 62
- LDAP, 50, 222
- LSA, 76

- Man-in-the-middle attack, 198, 229
- MCUs, 159
- MIB, 295
- MPPE, 229

- NAT, 112, 218
- NAT, 112
- NCPs, 62
- NTP, 98

- OSI, 7
- OSPF, 73, 74

- PAP, 62, 135, 229
- PBR, 88
- PFS, 217
- PPP, 27, 62, 135, 228
- PPPoE, 27, 61
- PPTP, 27, 228
- Proxy ARP, 94
- PSK, 216, 220

- QoS, 247, 250

- RADIUS, 135
- Replay attack, 229
- RIP, 72
- Route, 69
- RouteFailover, 77
- Router priority, 75
- Routing table, 69

- RSA, 197

- SA, 215
- SAT, 114, 115
- SNMP, 28, 294
- SNTP, 98
- SPF, 76
- Spoofing, 123
- SSL, 136, 243
- Stub area, 75
- SYN flooding, 47

- T.120, 160
- TLS, 136, 243
- ToS, 250
- Twofish, 196

- UDP/TIME, 98
- URL, 157

- VLAN, 56
- VLink, 74
- VoIP, 158
- VPN, 13, 193, 207

- WWW, 155

APPENDIX A

Console Commands Reference

This Appendix contains the list of commands that can be used in CLI for monitoring and troubleshooting the firewall. For information about how to access the CLI from a PC or terminal, please refer to [4.2, Monitoring Via CLI](#).

List of Commands

About

Brings up information pertaining to the version of the firewall core in use and a copyright notice.

- **Syntax:** about

Example:

```
Cmd> about
```

```
D-Link DFL 2.01.00V
```

```
Copyright Clavister 1996-2005. All rights reserved
```

```
SSH IPSEC Express SSHIPM version 5.1.1 library 5.1.1
```

```
Copyright 1997-2003 SSH Communications Security Corp.
```

```
Build : Jun 3 2005
```

Access

Displays the contents of the Access configuration section.

- **Syntax:** access

Example:

```
Cmd> access

Source IP Address  Access list (spoofing protection)
Rule Name          Action  Iface          Source Range
-----
If no access rule matches, PBR rules will be used to select
a routing table to be used for reverse route lookup.  If the
route lookup fails, the action will be DROP.
```

ARP

Displays ARP entries for the specified interface(s). Published, static as well as dynamic items are shown.

- **Syntax:** arp [options] <interface pattern>
- **Options:**
 - ip <pattern> –Display only IP addresses matching <pattern>
 - hw <pattern> –Display only hardware addresses matching <pattern>
 - num <n> –Display only the first <n> entries per iface (default: 20)
 - hashinfo –Display information on hash table health
 - flush –Flush ARP cache of ALL interfaces
 - flushif –Flush ARP cache of an iface

Example:

```
Cmd> arp wan

ARP cache of iface wan
Dynamic 194.2.1.1      = 0020:d216:5eec   Expire=141
```

ARPSnoop

Toggles the on-screen display of ARP queries. This command can be of great help in configuring firewall hardware, since it shows which IP addresses are heard on each interface.

- **Syntax:**

- arpsnoop <interface pattern>
Toggle snooping on given interfaces.
- arpsnoop all
Snoop all interfaces.
- arpsnoop none
Disable all snooping.

Example:

```
Cmd> arpsnoop all

ARP snooping active on interfaces:  lan wan dmz
ARP on wan:  gw-world requesting ip_ext
ARP on lan:  192.168.123.5 requesting lan_ip
...
```

Buffers

This command can be useful in troubleshooting; e.g. if an unexpectedly large number of packets begin queuing in the firewall or when traffic does not seem to be flowing for some inexplicable reason. By analyzing the contents of the buffers, it is possible to determine whether such traffic is making it to the firewall at all.

- **Syntax:**

- buffers
Brings up a list of most recently freed buffers.

Example:


```
Cmd> buffers
```

Displaying the 20 most recently freed buffers

RecvIf	Num	Size	Protocol	Sender	Destination
wan	1224	121	UDP	192.168.3.183	192.168.123.137
lan	837	131	UDP	192.168.123.137	192.168.3.183
wan	474	112	UDP	192.168.3.183	192.168.123.137
wan	395	91	UDP	192.168.3.183	192.168.123.137
...					

```
-- buffer <number>
```

Shows the contents of the specified buffer.

Example:

```
Cmd> buff 1059
```

Decode of buffer number 1059

lan:Enet 0050:dadf:7bbf->0003:325c:cc00, type 0x0800, len 1058

IP 192.168.123.10->193.13.79.1 IHL:20 DataLen:1024 TTL:254

Proto:ICMP

ICMP Echo reply ID:6666 Seq:0

```
-- buffer .
```

Shows the contents of the most recently used buffer.

Example:

```
Cmd> buff .
```

Decode of buffer number 1059

lan:Enet 0050:dadf:7bbf->0003:325c:cc00, type 0x0800, len 1058

IP 192.168.123.10->193.13.79.1 IHL:20 DataLen:1024 TTL:254

Proto:ICMP

ICMP Echo reply ID:6666 Seq:0

Certcache

Displays the contents of the certificate cache.

- **Syntax:** certcache

CfgLog

Shows the results of the most recent reconfiguration or start up of the firewall. This text is the same as is shown on-screen during reconfiguration or start up.

- **Syntax:** `cfglog`

Example:

```
Cmd> cfglog
```

```
Configuration log: License file successfully loaded.  
Configuration done
```

Connections

Shows the last 20 connections opened through the firewall. Connections are created when traffic is permitted to pass via Allow or NAT rules. Traffic permitted to pass under FwdFast is not included in this list.

Each connection has two timeout values, one in each direction. These are updated when the firewall receives packets from each end of the connection. The value shown in the Timeout column is the lower of the two values. Possible values in the State column include:

- SYN_RECV TCP packet with SYN flag received
- SYNACK_S TCP packet with SYN + ACK flags sent
- ACK_RECV TCP packet with ACK flag received
- TCP_OPEN TCP packet with ACK flag sent
- FIN_RECV TCP packet with FIN/RST flag received
- PING The connection is an ICMP ECHO connection
- UDP The connection is a UDP connection
- RAWIP The connection uses an IP protocol other than TCP, UDP or ICMP

- **Syntax:** `connections`

Example:

```
Cmd> conn
```

State	Proto	Source	Destination	Tmout
TCP_OPEN	TCP	wan:60.20.37.6:5432	dmz:wwsrv:80	3600
SYN_RECV	TCP	wan:60.20.37.6:5433	dmz:wwsrv:80	30
UDP_OPEN	UDP	lan:10.5.3.2:5433	dmz:dnsrv:53	50

Cpuid

Shows information regarding the CPU in the firewall hardware.

- **Syntax:** cpuid

Example:

```
Cmd> cpuid

Processor:           Intel Pentium 4
Est. frequency:     1402 MHz
Family:             15
Model:              0
Stepping:           10
Vendor ID:          GenuineIntel
Type:               Original OEM Processor
Logical CPUs (HTT): 1
Feature flags:      fpu vme de pse tsc msr pae mce cx8 apic
                   sep mtrr pge mca cmov pat pse-36 clfsh
                   ds acpi mmx fxsr sse sse2 ss htt tm
Cache and TLB information:
0x66: 1st-level data cache: 8-KB, 4-way set associative,
      sectored cache, 64-byte line size
...
```

DHCP

- **Syntax:** dhcp [options] <interface>
- **Options:**
 - renew – Force interface to renew it's lease
 - release – Force interface to release it's lease

Example:

```
Cmd> dhcp -renew wan
```

DHCPRelay

Show the contents of the DHCP-relay configuration section.

- **Syntax:** `dhcrelay [options]`
- **Options:**
 - `release ip` – Releases the IP and removes associated routes from the firewall.

Example:

```
Cmd> dhcrelay
```

DHCPServer

Show the contents of the DHCP-server configuration section and active DHCP leases.

- **Syntax:** `dhcserver [options]`
- **Options:**
 - `rules` – Shows dhcp server rules
 - `leases` – Shows dhcp server leases
 - `mappings` – Shows dhcp server IP→MAC mappings
 - `release` – Releases an active or blacklisted IP

Example:

```
Cmd> dhcserver
```

DynRoute

Displays the dynamic routing policy filter ruleset and current exports.

- **Syntax:** `dynroute [options]`
- **Options:**
 - `rules` – Display dynamic routing filter ruleset
 - `exports` – Display current exports

FragS

Shows the 20 most recent fragment reassembly attempts. This includes both ongoing and completed attempts.

- **Syntax:** frags

Example:

```
Cmd> frags
```

RecvIf	Num	State	Source	Dest.	Protocol	Next	Tout
-----	---	-----	-----	-----	-----	-----	-----
lan	2	Done	10.5.3.2	26.23.5.4	ICMP	2000	58
wan	8	Accept	23.3.8.4	10.5.3.2	ICMP	1480	60

HA

Shows information about a HA cluster.

- **Syntax:** ha

Example:

```
Cmd> ha

This device is a HA SLAVE
This device is currently ACTIVE (will forward traffic)
HA cluster peer is ALIVE
```

HTTPPoster

Show the configured httpposter urls and status.

- **Syntax:** httpposter [options]
- **Options:**
 - repost – Re-post all URLs now.

Example:

```
Cmd> httpposter

HTTPPoster_URL1:
Host : ""
Port : 0
Path : ""
Post : ""
User : ""
Pass : ""
Status: (not configured)
...
```

Ifacegroups

Shows the configured interface groups.

- **Syntax:** `ifacegroups <name pattern>`

Example:

```
Cmd> ifacegroups

Configured interface groups
-----
internals lan,vlan1,vlan2,vlan3
```

IfStat

- **Syntax:**
`-- ifstat`
Shows a list of the interfaces installed in the firewall.

Example:

```
Cmd> ifstat

Configured interfaces:
Interface name  IP Address      Interface type
-----
core           127.0.0.1      Null (sink)
wan            172.16.87.252  ...
lan           192.168.121.1  ...
```

```
-- ifstat <interface>
```

Shows hardware and software statistics for the specified NIC.

Example:

```
Cmd> ifstat lan

Iface lan
...
MTU :      ...
IP Address : ...
Hw Address : ...
Software Statistics:
Soft received: ...  Soft sent:      ...  Send failures: ...
Dropped:      ...  IP Input Errs:  ...
Driver information / hardware statistics:
...
```

The *Dropped* counter in the software section states the number of packets discarded as the result of structural integrity tests or firewall ruleset drops.

The *IP Input Errs* counter in the software section specifies the number of packets discarded due to checksum errors or IP headers broken beyond recognition. The latter is most likely the result of local network problems rather than remote attacks.

Ikesnoop

Ikesnoop is used to diagnose problems with IPsec tunnels.

- **Syntax:**

```
-- ikesnoop
```

Display current ikesnoop status.

```
-- ikesnoop off
```

Turn IKE snooping off.

```
-- ikesnoop on [ipaddr]
```

Turn IKE snooping on, if a IP is specified only ike traffic from that IP will be showed.

```
-- ikesnoop verbose [ipaddr]
```

Enable verbose output, if a IP is specified only ike traffic from that IP will be showed.

Ipseckeealive

Shows the status of the configured Ipsec keepalive connections.

- **Syntax:** ipseckeealive

Example:

```
Cmd> ipseckeealive

192.168.0.10 -> 192.168.1.10: Consecutive lost: 0, sent:
908, lost: 2
192.168.1.10 -> 192.168.0.10: Consecutive lost: 0, sent:
913, lost: 6
```

IPSec tunnels

Display configured IPSec VPN connections.

- **Syntax:** ipsectunnels

Example:

```
Cmd> ipsectunnel

No  Name      Local Net      Remote Net  Remote GW
0   vpn-home  192.168.123.0/24  0.0.0.0/0  None
MAIN_MODE SA_PER_NET DONT_VERIFY_PAD IKE group: 2
IKE proplist:  ike-default, IPsec proplist:
esp-tn-roamingclients
```

IPSecstats

Display connected IPSec VPN gateways and remote clients.

- **Syntax:** ipsecstats [options]
- **Options:**
 - ike Displays IKE SAs
 - ipsec Displays IPsec SAs (default)
 - u Displays detailed SA statistic information
 - v Displays verbose information

- num <n> Maximum number of entries to display (default: 40/8)
Note: if set to 0, ALL entries will be displayed

Example:

```
Cmd> ipsecstats

--- IPsec SAs:
Displaying one line per SA-bundle
...
```

Killsa

Kills all IPsec and IKE SAs for the specified IP-address.

- **Syntax:** killsa <ipaddr>

Example:

```
Cmd> killsa 192.168.0.2

Destroying all IPsec & IKE SAs for remote peer 192.168.0.2
```

License

Shows the content of the license-file. It is also possible to remove a license from a running firewall with this command, by doing a license remove.

- **Syntax:** license [remove]

Example:

```
Cmd> lic

Contents of the License file
-----
Registration key:      ...
Bound to MAC address: ...
Model:                DFL-...
Registration date:    ...
Issued date:          ...
Last modified:        ...
New upgrades until:   ...

Ethernet Interfaces:  ...
...
```

Lockdown

Sets local lockdown on or off. During local lockdown, only traffic from admin nets to the firewall itself is allowed. Everything else is dropped.

Note: If local lockdown has been set by the core itself due to licensing or configuration problems, this command will NOT remove such a lock.

- **Syntax:** lockdown [on | off]

Loghosts

Shows the list of log recipients the firewall is configured to send log data to.

- **Syntax:** loghosts

Example:

```
Cmd> loghosts

Log hosts:
SysLog 192.168.123.10  Facility: local0
Usage logging in 3600 second intervals
```

Memory

Displays core memory consumption. Also displays detailed memory use of some components and lists.

- **Syntax:** memory

Netcon

Shows a list of users currently connected to the firewall via the netcon management protocol.

- **Syntax:** netcon

Example:

```
Cmd> netcon

Currently connected NetCon users:
Iface IP address      port
lan    192.168.123.11    39495
```

Netobjects

Displays the list of named network objects and their contents. If a netobject is specified the output will show user authentication information associated with that object.

- **Syntax:** `netobjects [options]`
- **Options:**
 - `num <number>` maximum objects listed (default: 20)
 - `dump` make netobject dump MUCH more detailed (debug_cmds)

OSPF

Shows runtime information about the OSPF router processes) and is used to stop/start OSPF processes).

- **Syntax:** `ospf [<process name>] [<parameter>] [<argument>]`
- **Available parameters:**
 - `iface [<iface>]`, Display interface information
 - `area [<areaID>]`, Display area information
 - `neighbor [<if>:][<neiID>]`, Display neighbor information
 - `route`, Display the internal OSPF process routingtable
 - `database [verbose]`, Display the LSA database
 - `lsa <lsaID>`, Display details for a specified LSA
 - `snoop [on | off]`, Display troubleshooting messages on the console
 - `ifacedown <iface>`, Takes specified interface offline
 - `ifaceup <iface>`, Takes specified interface online
 - `stop`, Stop OSPF process
 - `start`, Start OSPF process
 - `restart`, Restart OSPF process

- **Debug parameters:**

- spf, Performs full SPF calculation
- refresh, Refreshes all self originated LSAs in the process
- ifacemetric <if> <metric>, Changes the metric of a interface

Ping

Sends a specified number of ICMP Echo Request packets to a given destination. All packets are sent in immediate succession rather than one per second. This behavior is the best one suited for diagnosing connectivity problems.

- **Syntax:** ping <IPAddr> [options] [<# of packets> [<size>]]

- **Options:**

- r <recvif>, Run through the ruleset, simulating that the packet was received by <recvif>.
- s <srcip>, Use this source IP.
- p <table>, Route using the specified PBR table.
- v, Verbose ping.

Example:

```
Cmd> ping 192.168.12.1
```

```
Sending 1 ping to 192.168.12.1 from 192.168.14.19 using PBR  
table "main". Echo reply from 192.168.12.1 seq=0 time= 10 ms  
TTL=255
```

Pipes

Shows the list of configured pipes; the contents of the Pipes configuration section, along with basic throughput figures of each pipe.

- **Syntax:** pipes [options] <name>

- **Options:**

- s Display overall statistics

- u Display users of a given pipe <name>

Example:

```
Cmd> pipes

Configured pipes:
Name      Grouping      Bits/s  Pkts/s  Precedence
-----  -
std-in    Per DestIP                    0 1 7
Current:  42.5 K 21.0
...
```

Proplists

Lists the configured proposal lists.

- **Syntax:** proplists [vpnconn]

Example:

```
Cmd> propl

Displaying all configured proposal lists:
ike-default
...
```

ReConfigure

Re-reads the FWCore.cfg file from disk. This process takes approximately one second if done from floppy disk, and approximately a tenth of a second from hard disk or flash disk. If there is a FWCore_N.cfg file present on the disk, this will be read instead. However, as there is no Firewall Manager to attempt two-way communication with the firewall, it will conclude that the configuration is incorrect and revert to FWCore.cfg after the bi-directional verification timeout period has expired (typically 30 seconds).

- **Syntax:** reconfiure

Example:

```
Cmd> reconfigure

Shutdown RECONFIGURE. Active in 1 seconds.
Shutdown reason: Reconfigure due to console command
```

Remotes

Shows the contents of the Remotes configuration section.

- **Syntax:** `remotes`

Example:

```
Cmd> remotes
```

```
Hosts/nets with remote control of firewall:
```

```
...
```

```
WebUI HTTP (port 80) and HTTPS (port 443) access
```

Routes

Displays information about the routing tables, contents of a (named) routing table or a list of routing tables, along with a total count of route entries in each table, as well as how many of the entries are single-host routes. Note that "core" routes for interface IP addresses are not normally shown, use the "-all" switch to show core routes also.

In the "Flags" field of the routing tables, the following letters are used:

- O: Learned via OSPF X: Route is Disabled
- M: Route is Monitored A: Published via Proxy ARP
- D: Dynamic (from e.g. DHCP relay, IPsec, L2TP/PPP servers, etc.)

- **Syntax:** `routes [options] <table name>`

- **Options:**

- all, Also show routes for interface addresses
- num <n>, Limit display to <n> entries (default: 20)
- nonhost, Do not show single-host routes
- tables, Display list of named (PBR) routing tables
- lookup <ip>, Lookup the route for the given IP address
- v, Verbose

Rules

Shows the contents of the Rules configuration section.

- **Syntax:** `rules [options] <range>`
The range parameter specifies which rules to include in the output of this command.
- **Options:**
 - r, Show policy based routing ruleset
 - p, Show pipe ruleset
 - i, Show intrusion detection ruleset
 - t, Show threshold ruleset
 - v, Be verbose: show all parameters of the rules
 - s, Filter out rules that are not currently allowed by selected schedules

Example:

```
Cmd> rules -v 1

Contents of ruleset; default action is DROP
#  Act.  Source          Destination      Protocol/Ports
--  -
1  Allow  lan:  ...        core:  ...        "HTTP"
    "HTTP-fw" Use:  0 FWLOG:notice SYSLOG:notice
```

Scrsave

Activates the screensaver included with the firewall core.

- **Syntax:** `scrsave`

Example:

```
Cmd> scr

Activating screen saver...
```

Services

Displays the list of named services. Services implicitly defined inside rules are not displayed.

- **Syntax:** `services [name or wildcard]`

Example:

```
Cmd> services

Configured services:
HTTP      TCP      ALL > 80
```

Shutdown

Instructs the firewall to perform a shutdown in a given number of seconds. It is not necessary to perform a shutdown before the firewall is powered off, as it does not keep any open files while running.

- **Syntax:** `shutdown <seconds>`
`-- Shutdown in <n> seconds (default: 5)`

Sysmsgs

Show the contents of the OS sysmsg buffer.

- **Syntax:** `sysmsgs`

Example:

```
Cmd> sysmsg

Contents of OS sysmsg buffer:
...
```

Settings

Shows the contents of the Settings configuration section.

- **Syntax:**
`-- settings` Shows available groups of settings.

Example:

```
Cmd> sett
Available categories in the Settings section:
IP          - IP (Internet Protocol) Settings
TCP         - TCP (Transmission Control Protocol) Settings
ICMP        - ICMP (Internet Control Message Protocol)
ARP         - ARP (Address Resolution Protocol) Settings
State       - Stateful Inspection Settings
ConnTimeouts - Default Connection timeouts
LengthLim   - Default Length limits on Sub-IP Protocols
Frag        - Pseudo Fragment Reassembly settings
LocalReass  - Local Fragment Reassembly Settings
VLAN        - VLAN Settings
SNMP        - SNMP Settings
DHCPClient  - DHCP (Dynamic Host Configuration Protocol)
              Client Settings
DHCPRelay   - DHCP/BOOTP Relaying Settings
DHCPServer  - DHCP Server Settings
IPsec       - IPsec and IKE Settings
Log         - Log Settings
SSL         - SSL Settings
HA          - High Availability Settings
Timesync    - Time Synchronization Settings
DNSClient   - DNS Client Settings
RemoteAdmin - Settings regarding remote administration
Transparency - Settings related to transparent mode
HTTPPoster  - Post user-defined URLs periodically
              for e.g. dyndns registration, etc
WWWSrv      - Settings regarding the builtin web server
HwPerformance - Hardware performance parameters
IfaceMon    - Interface Monitor
RouteFailOver - Route Fail Over Default values
IDS         - Intrusion Detection / Prevention Settings
PPP         - PPP (L2TP/PPTP/PPPoE) Settings
Misc        - Miscellaneous Settings
```

```
-- settings <group_name>
Shows the settings of the specified group.
```

Example:

```
Cmd> settings arp

ARP (Address Resolution Protocol) Settings
  ARPMatchEnetSender      : DropLog
  ARPQueryNoSenderIP     : DropLog
  ARPSenderIP            : Validate
  UnsolicitedARPReplies  : DropLog
  ARPRequests            : Drop
  ARPChanges              : AcceptLog
  StaticARPChanges       : DropLog
  ARPExpire               : 900 ARPExpireUnknown : 3
  ARPMulticast           : DropLog
  ARPBroadcast           : DropLog
  ARPCacheSize           : 4096 ARPHashSize : 512
  ARPHashSizeVLAN        : 64
```

Stats

Shows various vital stats and counters.

- **Syntax:** stats

Example:

```
Cmd> stats
Uptime           : ...
Last shutdown    : ...
CPU Load         : 6
Connections      : 4919 out of 32768
Fragments        : 17 out of 1024 (17 lingering)
Buffers allocated : 1252
Buffers memory   : 1252 x 2292 = 2802 KB
Fragbufs allocated : 16
Fragbufs memory  : 16 x 10040 = 156 KB
Out-of-buffers   : 0
ARP one-shot cache : Hits : 409979144 Misses : 186865338
Interfaces: Phys:2 VLAN:5 VPN:0
Access entries:18 Rule entries:75
Using configuration file "FWCore.cfg", ver ...
```

Time

Displays the system date and time

- **Syntax:** `time [options]`
- **Options:**
 - `set <arg>`, Set system local time (YYYY-MM-DD HH:MM:SS)
 - `sync`, Synchronize time with timeserver(s) (specified in settings)
 - `force`, Force synchronization regardless of the MaxAdjust setting

Uarules

Displays the contents of the User Authentication ruleset.

- **Syntax:** `uarules [options] <range>`
- **Options:**
 - `v`, (verbose)show all parameters of the user authentication rules

Example:

```
Cmd> uarules -v 1-2
```

```
Contents of the User Authentication ruleset
```

#	Source Net	Agent	Auth source	Auth. Server
1	if1:192.168.0.0/24	HTTPAuth	RADIUS	FreeRadius
2	*:0.0.0.0/0	XAuth	RADIUS	IASRadius

Userauth

Displays currently logged-on users and other information. Also allows logged-on users to be forcibly logged out.

- **Syntax:** `userauth [options]`
- **Options:**
 - `l`, displays a list of all authenticated users
 - `p`, displays a list of all known privileges (usernames and groups)

- v <ip>, displays all known info for user(s) with this IP
- r <ip> <interface>, forcibly logs out an authenticated user
- num <num>, maximum number of authenticated users to list (default 20)

Example:

```

Cmd> userauth -l

Currently authenticated users:
Login      IP Address      Source      Ses/Idle      Privileges
-----      -
user1      ...             ...             1799          members
...

```

Userdb

Lists user databases and their contents.

- **Syntax:** userdb <dbname> [<wildcard> or <username>]
If <dbname> is specified users configured in that user database will be shown. A wildcard can be used to only show users matching that pattern or if a username is specified information regarding that user will be shown.
- **Options:**
 - num, Displays the specified number of users (default 20)

Example:

```

Cmd> userdb

Configured user databases:
Name      #users
-----      -
AdminUsers  1

```

Example:

```
Cmd> userdb AdminUsers
```

```
Configured user databases:
```

Username	Groups	Static IP	Remote Networks
admin	administrators		

Example:

```
Cmd> userdb AdminUsers admin
```

```
Information for admin in database AdminUsers:
```

```
Username : admin
Groups   : administrators
Networks :
```

Vlan

Shows information about configured VLANs.

- **Syntax:**
-- vlan
List attached VLANs

Example:

```
Cmd> vlan
```

```
VLANs:
```

```
vlan1 IPAddr: 192.168.123.1 ID: 1 Iface: lan
vlan2 IPAddr: 192.168.123.1 ID: 2 Iface: lan
vlan3 IPAddr: 192.168.123.1 ID: 3 Iface: lan
```

-- vlan <vlan>

Show information about specified VLAN.

Example:

```
Cmd> vlan vlan1
```

```
VLAN vlan1
```

```
Iface lan, VLAN ID: 1
```

```
Iface      : lan
```

```
IP Address  : 192.168.123.1
```

```
Hw Address  : 0003:474e:25f9
```

```
Software Statistics:
```

```
Soft received : 0 Soft sent: 0 Send failures: 0
```

```
Dropped      : 0 IP Input Errs : 0
```


APPENDIX **B**

Customer Support

International Offices

U.S.A

17595 Mt. Herrmann Street
Fountain Valley, CA 92708
TEL: 1-800-326-1688
URL: www.dlink.com

Canada

2180 Winston Park Drive
Oakville, Ontario, L6H 5W1
Canada
TEL: 1-905-8295033
FAX: 1-905-8295223
URL: www.dlink.ca

Europe (U. K.)

4th Floor, Merit House
Edgware Road, Colindale
London NW9 5AB
U.K.
TEL: 44-20-8731-5555
FAX: 44-20-8731-5511
URL: www.dlink.co.uk

Germany

Schwalbacher Strasse 74
D-65760 Eschborn
Germany
TEL: 49-6196-77990
FAX: 49-6196-7799300
URL: www.dlink.de

France

No.2 allée de la Fresnerie
78330 Fontenay le Fleury
France
TEL: 33-1-30238688
FAX: 33-1-30238689
URL: www.dlink.fr

Netherlands

Weena 290
3012 NJ, Rotterdam
Netherlands
Tel: +31-10-282-1445
Fax: +31-10-282-1331
URL: www.dlink.nl

Belgium

Rue des Colonies 11
B-1000 Brussels
Belgium
Tel: +32(0)2 517 7111
Fax: +32(0)2 517 6500
URL: www.dlink.be

Italy

Via Nino Bonnet n. 6/b
20154 Milano
Italy
TEL: 39-02-2900-0676
FAX: 39-02-2900-1723
URL: www.dlink.it

Sweden

P.O. Box 15036,
S-167 15 Bromma
Sweden
TEL: 46-(0)8564-61900
FAX: 46-(0)8564-61901
URL: www.dlink.se

Denmark

Naverland 2,
DK-2600 Glostrup,
Copenhagen
Denmark
TEL: 45-43-969040
FAX: 45-43-424347
URL: www.dlink.dk

Norway

Karihaugveien 89
N-1086 Oslo
Norway
TEL: +47 99 300 100
FAX: +47 22 30 95 80
URL: www.dlink.no

Finland

Latokartanontie 7A
FIN-00700 Helsinki
Finland
TEL: +358-10 309 8840
FAX: +358-10 309 8841
URL: www.dlink.fi

Spain

C/Sabino De Arana
56 Bajos
08028 Barcelona
Spain
TEL: 34 93 4090770
FAX: 34 93 4910795
URL: www.dlink.es

Portugal

Rua Fernando Pahlá
50 Edificio Simol
1900 Lisbon
Portugal
TEL: +351 21 8688493
URL: www.dlink.es

Czech Republic

Vaclavske namesti 36, Praha 1
Czech Republic
TEL :+420 (603) 276 589
URL: www.dlink.cz

Switzerland

Glatt Tower, 2.OG CH-8301
Glattzentrum Postfach 2.OG
Switzerland
TEL : +41 (0) 1 832 11 00
FAX: +41 (0) 1 832 11 01
URL: www.dlink.ch

Greece

101, Panagoulis Str. 163-43
Helioupolis Athens,
Greece
TEL : +30 210 9914 512
FAX: +30 210 9916902
URL: www.dlink.gr

Luxemburg

Rue des Colonies 11,
B-1000 Brussels,
Belgium
TEL: +32 (0)2 517 7111
FAX: +32 (0)2 517 6500
URL: www.dlink.be

Poland

Budynek Aurum ul. Walic-w 11
PL-00-851 Warszawa
Poland
TEL : +48 (0) 22 583 92 75
FAX: +48 (0) 22 583 92 76
URL: www.dlink.pl

Hungary

R-k-czi-t 70-72
HU-1074 Budapest
Hungary
TEL : +36 (0) 1 461 30 00
FAX: +36 (0) 1 461 30 09
URL: www.dlink.hu

Singapore

1 International Business Park
#03-12The Synergy
Singapore 609917
TEL: 65-6774-6233
FAX: 65-6774-6322
URL: www.dlink-intl.com

Australia

1 Giffnock Avenue
North Ryde, NSW 2113
Australia
TEL: 61-2-8899-1800
FAX: 61-2-8899-1868
URL: www.dlink.com.au

India

D-Link House, Kurla Bandra
Complex Road OffCST Road,
Santacruz (East)
Mumbai - 400098
India
TEL: 91-022-26526696/56902210
FAX: 91-022-26528914
URL: www.dlink.co.in

Middle East (Dubai)

P.O.Box: 500376
Office: 103, Building:3
Dubai Internet City
Dubai, United Arab Emirates
Tel: +971-4-3916480
Fax: +971-4-3908881
URL: www.dlink-me.com

Turkey

Ayazaga Maslak Yolu
Erdebil Cevahir Is Merkezi
5/A Ayazaga Istanbul
Turkiye
TEL: +90 212 289 56 59
FAX: +90 212 289 76 06
URL: www.dlink.com.tr

Egypt

19 El-Shahed Helmy, El Masri
Al-Maza, Heliopolis
Cairo, Egypt
TEL:+202 414 4295
FAX:+202 415 6704
URL: www.dlink-me.com

Israel

11 Hamanofim Street
Ackerstein Towers,
Regus Business Center P.O.B 2148,
Hertzelia-Pituach 46120
Israel
TEL: +972-9-9715700
FAX: +972-9-9715601
URL: www.dlink.co.il

LatinAmerica

Isidora Goyechea 2934
Ofcina 702
Las Condes
Santiago Chile
TEL: 56-2-232-3185
FAX: 56-2-232-0923
URL: www.dlink.cl

Brazil

Av das Nacoes Unidas
11857 14- andar - cj 141/142
Brooklin Novo
Sao Paulo - SP - Brazil
CEP 04578-000 (Zip Code)
TEL: (55 11) 21859300
FAX: (55 11) 21859322
URL: www.dlinkbrasil.com.br

South Africa

Einstein Park II
Block B
102-106 Witch-Hazel Avenue
Highveld Technopark
Centurion
Gauteng Republic of South Africa
TEL: 27-12-665-2165
FAX: 27-12-665-2186
URL: www.d-link.co.za

Russia

Grafsky per., 14, floor 6
Moscow
129626 Russia
TEL: 7-095-744-0099
FAX: 7-095-744-0099 #350
URL: www.dlink.ru

China

No.202,C1 Building,
Huitong Office Park,
No. 71, Jianguo Road,
Chaoyang District,
Beijing 100025, China
TEL +86-10-58635800
FAX: +86-10-58635799
URL: www.dlink.com.cn

Taiwan

2F, No. 119, Pao-Chung Rd.
Hsin-Tien, Taipei
Taiwan
TEL: 886-2-2910-2626
FAX: 886-2-2910-1515
URL: www.dlinktw.com.tw

Headquarters

2F, No. 233-2, Pao-Chiao Rd.
Hsin-Tien, Taipei
Taiwan
TEL: 886-2-2916-1600
FAX: 886-2-2914-6299
URL: www.dlink.com.tw

