

# Linux Exposed

## How Proxies Work

Date Monday, April 05 @ 12:07:01

Topic Internet

Proxy servers were originally developed to cache frequently accessed web pages for computers behind a common Internet connection. In the early days of the Internet, wide area links were very slow, the Web was relatively small, and web pages were static. The entire Web consisted of only a few thousand websites shared by scientists and academicians. Whenever an important new element hit a website, many scientists in the same organization would visit that page (how many times have you forwarded a link inside your company?). By caching that page on a local server, proxies could eliminate redundant Internet access to retrieve the same page over and over. So, proxies were originally very effective at web caching.

When the Web went supernova, proxies became markedly less effective at caching; the Web was now vast, web pages were frequently dynamic (expiring as soon as they'd been transmitted), and the interests of users within a single organization might range across a million web pages before the same site was hit three times. These factors presented a difficult caching problem indeed and proxies became largely ineffective, except in extremely large organizations or in ISPs. Although support for proxy servers was built into all the standard browsers, by 1996 it was seldom used.

But the new Web also has its seedier element, and proxy servers showed a remarkably serendipitous side effect: They can hide all the real users of a network behind a single machine, they can filter URLs, and they can drop suspicious or illegal content. So although originally created as non-security caches, the primary purpose of the majority of proxy servers has now become firewalled.

Proxy servers regenerate high-level service requests on an external network on behalf of their clients on a private network. This effectively hides the identity and number of clients on the internal network from examination by the external network. Because of their position between a number of internal clients and public servers, proxies can also cache frequently accessed content from the public network to reduce access to the public network through high-cost wide-area links.

For the sake of understanding, this article discusses only "pure" proxies those that operate on the principle of service protocol forwarding. Most actual implementations of security proxies include the services of packet filtering and Network Address Translation to form a complete firewall. Those technologies can be combined with proxies to eliminate some of the attacks to which pure proxies are vulnerable.

Many proxy service alternatives exist, ranging from the Application layer filter functionality of true firewalls like Checkpoint's Firewall-1, to general-purpose pure "proxy only" applications like WinGate, to simple single-service proxies like Jigsaw for HTTP. Pure proxies are subject to a number of problems, most based on the fact that the base operating system is not protected by the proxy software against denial-of-service attacks and the exploitation of other services that may be running on the server.

Proxy servers are most often associated with the HTTP World Wide Web service because proxies were first developed for this service. Since that time, proxy functionality has been applied to most other common Internet

services. Examples in this article will use the HTTP service, but the functionality remains largely the same for other services.

## How Proxies Work

Proxies work by listening for service requests from internal clients and then sending those requests on the external network as if the proxy server itself were the originating client. When the proxy server receives a response from the public server, it returns that response to the original internal client as if it were the originating public server.

## Security Advantages of Proxies

The process of request regeneration and the fact of a proxy's location between the external and internal networks provide a number of security advantages:

1. Proxies hide private clients from external exposure.
2. Proxies can block dangerous URLs.
3. Proxies can filter dangerous content such as viruses and Trojan horses before passing it to the client.
4. Proxies can check the consistency of returned content.
5. Proxies can eliminate the need for transport layer routing between networks.
6. Proxies provide a single point of access, control, and logging.

Each of these security advantages is detailed in the following sections.

### *Client Hiding*

The major security feature of proxy servers is client hiding. Like Network Address Translation, proxy servers can make an entire internal network appear to be a single machine from the Internet because only a single machine passes requests onto the Internet.

Like Network Address Translators, proxy servers prevent external hosts from connecting to services on internal machines. In the case of proxy servers, no route to the clients exists because the address domains of the internal and external networks may be incompatible and because transport layer routing does not exist between the two networks.

Proxies perform this feature by completely regenerating service-level requests rather than simply changing and recalculating address headers. For example, when a web client makes a request through a proxy server, the proxy server receives the request as if it were the destination web server on the internal network. It then regenerates the request on the external network as if it were a typical web browser. When the proxy receives the response from the ultimate web server, it serves that response to its internal client. Only HTTP passes through the proxy, not TCP or IP. TCP/IP (and other low-level protocols) are regenerated by the proxy; they do not route through it unless the proxy is misconfigured.

Another aspect of client hiding is that of connection multiplexing; a proxy server can be used to share a single Internet connection and IP address among an entire network. For this reason, lightweight proxy servers like WinGate are extremely popular in home and small office environments where only a single dial-up or dedicated connection is available.

### *URL Blocking*

URL blocking allows administrators to disallow the provision of certain websites based on their URLs. In theory, this will keep your employees from viewing websites you don't want them to have access to. This function in

easy to implement. The proxy simply checks every request for a webpage (or other service URL) against a list of denied pages before it regenerates the request. If the URL is blocked, the proxy will not request or return the page.

URL blocking is easy to circumvent, however, because a website can be just as easily addressed by its IP address or even by the whole number address. For example, a user could type in any of the following in their web browser to access exactly the same home page:

*<http://www.linuxexposed.com/index.php>*

*<http://212.190.116.128/index.php>*

*<http://59672698250359936/index.php>*

But your URL blocker will (probably) only be checking for the full text URL.

URLs can contain DNS names or IP addresses. Most people are familiar with the first two examples of site references, but have never heard of the third: an IP address specified as a whole number rather than as a "dotted quad notation." The concept is simple: An IP address is just a 32-bit number, and though we refer to them in dotted quad (10.0.0.0) notation for convenience sake, there's no reason why they can't be referred to as whole numbers. To convert a dotted quad number to a whole number, use the following formula ("a" is the most significant quad, "d" the least):  $a \times 224 + b \times 216 + c \times 28 + d$ . Converting everything to easily calculable numbers, the formula becomes:  $a \times 16777216 + b \times 65536 + c \times 256 + d$ . So, for example, turning the IP address for <http://www.linuxexposed.com>, 209.68.11.152, into a whole number makes it  $209 \times 16777216 + 68 \times 65536 + 11 \times 256 + 152 = 59672698250359936$ . Put 59672698250359936 into your web browser's address bar and you'll see the Linux Exposed web page come up. Note that websites behind proxy servers (like Microsoft.com) don't come up because the whole number IP address must be programmed into the proxy for the proxy to recognize it.

The other major problem with URL blocking for security administrators is simply keeping up with sites to block. Problem sites like hacking depositories, pornographic sites, and game sites have the ephemeral life of a mayfly; they pop up and disappear just as quickly. Most people who engage in the activities ascribed by these sites just use search engines or Usenet news lists to keep up with where their favorite sites have moved. You will not be able to stay ahead of that activity with your URL-blocked database.

### ***Content Filtering***

Because proxies retransmit all protocol payloads and are protocol specific, the proxy service can be used to search the payload for suspicious content. This means that you can configure your HTTP proxy service to strip out ActiveX controls, Java applets, or even large images if you feel they could present a security problem. You could also use an SMTP proxy to strip out executable file attachments and archived zip files if you felt they were a security problem.

Content filters can also be used to check web pages for the presence of certain words or phrases, such as the trademarks of your competition or some current news item.

You should filter ActiveX controls in websites, Java applets, and executable files in e-mail because they can be used to install Trojan horses inside your network. If someone needs to transfer an executable file, have him or her transmit it as a zip file or use BinHex or some other encoder to transfer it in a text format. This will require effort to decode, thus preventing the accidental transfer of a virus or Trojan horse into your network.

### ***Consistency Checking***

Consistency checking refers to checking the content of a protocol to be sure it makes sense for that protocol. Consistency checking ensures that specifically malformed types of content can't be used to exploit a security weakness in your internal network.

For example, earlier versions of the Unix Sendmail SMTP daemon were notoriously susceptible to various buffer overflow problems. These occurred when an e-mail message was sent and it was longer than it said it was. Sendmail would allocate a memory buffer that was the size the message claimed to be, but would then scan the message until it hit an end-of-file marker. If the area between the claimed end and the actual end contained executable code, a hacker could gain root access to your e-mail server.

Another example of a buffer overflow occurs in URLs that are longer than 256 characters. Early web browsers were flawed because the end of the URL beyond 256 characters could contain executable code that would be executed by the browser software.

Of course by now everyone has heard of the amazing number of buffer overrun exploits that hackers have been able to perpetrate against IIS4 and IIS5. Using URLs and posts that are longer than IIS can handle, as well as exploits against supporting DLLs like the text search and legacy database support modules of IIS, hackers have been able to create completely automated attacks against IIS that operate as worms on the Internet and cause widespread havoc. Microsoft has released hundreds of security hotfixes to try to cope with the problem, and it has seriously impacted the marketability of their .NET services, which are all based on a web server that nobody wants to deploy because of its security problems.

Consistency checking with your proxy software can ensure that these sorts of problems are eliminated at the proxy so they won't affect internal machines. Unfortunately, the problems to check for usually are not known until some hacker exploits them, so most consistency checks are only available after an exploit has been found. And with automated worms, a large portion of the web servers on the net can be exploited within a few hours, so the "countermeasure" aspect of hotfixing servers is rather ineffective.

### ***Route Blocking***

Transport layer packets need not be routed because the request is completely regenerated. This eliminates Transport layer exploits like source routing, fragmentation, and various denial-of-service attacks. By eliminating routing, you can also ensure that any protocol for which you have not established a proxy service cannot be passed to the public network.

Route blocking is perhaps the most important advantage of proxy servers. Because no TCP/IP packets actually pass between the internal and external networks, a vast number of denial-of-service and exploitation attacks are prevented.

Unfortunately, route blocking is not used often enough. Because many protocols exist for which there are no good proxy services, administrators often must enable routing on the proxy server, which completely eliminates the security gain achieved by route disconnection. If you can, avoid allowing low-level network packets to pass through your proxy server. Most proxy server software will allow you to create generic TCP proxy services for any port using a generic SOCKS proxy or the Unix `redir` utility. These generic proxies, although they cannot perform content filtering, still allow you to keep TCP/IP packets from flowing between your networks.

### ***Logging and Alerting***

The final security advantage of proxies is the logging and alerting facilities they provide. Proxies ensure that all content flows through a single point, which gives you a checkpoint for network data. Most proxy software will log

the usage characteristics of the proxy by user and can be configured to retain a log of sites they visit. This will allow you to reconstruct the user's web browsing sessions if you suspect some illegal or unethical activity has occurred.

The alerting facility provided by some proxies can alert you to attacks in progress, even though the proxy facility of a server is not generally subject to attack. But the facility can alert you to attempted proxy connections from the external interface, which hackers frequently try to exploit to launder their connections.

## Performance Aspects of Proxies

In addition to their security aspects, proxy servers can also perform important performance enhancements:

1. Proxies can cache frequently requested data to improve performance by eliminating redundant access to the slower external network.
2. Proxies can balance the service load across a number of internal servers.

## Caching

As we noted in the beginning of this article, proxies were originally developed as a performance improvement, not as a security device. In the early Web days, there were only tens of thousands of sites. They were mostly scientific in nature and didn't change often. Because wide area links to the Internet were slow, a proxy could be used to cache large portions of the Web locally, so internal users could simply browse from the local proxy. Content didn't change very fast, so that made sense.

Caching now only makes sense in those instances where a large number of users frequently access the same web pages over and over. This usage pattern is currently somewhat rare, so the caching aspects of proxy servers are all but obsolete.

As e-commerce becomes more prevalent, caching will again become an important function because many people will perform their jobs by interfacing to a few frequently accessed sites. Consider for example a travel agency that uses Expedia.com and Travelocity.com to perform their work. Many agents would access the same two sites over and over, so caching the main site elements, graphics, and applets would make sense.

## Reverse Proxy Load Balancing

Newer proxy servers can be used to "reverse proxy," or provide the proxy service to external clients for internal servers. This functionality is used to balance the load of clients across a number of web servers. Many high functionality websites make use of complex applications in the form of ISAPI applications, Active Server Pages, Java servlets, or CGI applications. These applications execute on the server, so they considerably reduce the number of clients a single server can handle. For example, an NT server running IIS that could reasonably handle 100,000 browsers of standard HTML pages may only be able to handle 5,000 browsers of an ASP page that is executed on the server.

This means that most e-commerce functions cannot actually be handled on a single server, so the site must be run in parallel across a number of machines. For example, <http://www.microsoft.com/> is currently run on 30 identical web servers. DNS provides a rudimentary load-sharing scheme by which subsequent access to a DNS name will provide one of a number of IP addresses, but this does not actually balance the load. Only after a statistically large number of equivalent accesses occurs does this scheme serve to actually balance the client load.

A proxy server can be used to respond to a single IP address and then funnel client connections to one of a

number of site servers behind it. The proxy server can use some measurement provided by each web server to maintain awareness of which server has the most remaining capacity. Each connecting client can then be funneled to whichever server has the most capacity to handle it. Because the proxy actually does very little work compared to the effort of serving e-commerce webpages.

### ***Security Liabilities of Proxies***

Proxies suffer from some of the following security liabilities:

1. Proxies create a single point of failure.
2. Client software often must be capable of working with proxies. Only advanced firewall and proxy systems can be configured to work transparently on the network.
3. Proxies must exist for each service.
4. Proxies do not protect the base operating system.
5. Default configurations are often optimized for performance rather than security.

### ***Single Point of Failure***

Inherent with any single point of control is a single point of failure. If a hacker can disable your proxy, your entire organization could be cut off from the Internet.

Proxies, routers, and firewalls all suffer from this problem to some degree. With routers the problem is easily fixed by simply having more than one route to the Internet. Firewalls are far more secure than pure proxies because they include low-level packet filtering to eliminate the problems caused by denial-of-service activities. Pure proxy servers do not include the functionality to protect themselves from attack however, so they are very vulnerable both to intrusion and denial of service.

Modern proxy servers usually include a hot-failover feature where a secondary proxy with the same network connections constantly queries the "live" proxy and takes its IP addresses if it appears to have failed. Others use a load-balancing feature to provide multiple peer proxies that are all in use at the same time. The Windows Load Balancing feature of Windows 2000 Advanced Server can be configured with proxy server software to create this type of fault-tolerant proxy.

### ***Clients Must Be Made to Work with Proxies***

A proxy-enabled client must exist for each service you wish to proxy. For example, your web browser must support connection to a proxy server by including in the configuration options regarding which proxy service all requests should transmit to. If the client software cannot be configured to use a proxy, a proxy service cannot be used except in conjunction with a true Network Address Translator. This can be a major problem for services like FTP where the client software that ships with most operating systems does not support connection to a proxy server. You can purchase proxy clients for these services, however.

Proxy services included with address translating firewalls can get around this restriction because they can modify inbound and outbound network addresses. This means that clients need not know or be configured to work with proxies that exist as part of a true address translating firewall.

### ***Proxies Must Exist for Each Service***

A different proxy service is required for each supported service protocol. Network Address Translation is universal and works with any protocol except those that rely upon payload-embedded IP address information or require the ability to open a back channel to the client. Protocols for which no proxy service is available cannot

be connected through a proxy except by a generic TCP proxy service (like the generic SOCKS proxy) that would work much like a Network Address Translator. Any such service would not have the advantage of content filtering, however.

Many services cannot be easily proxied because they require the establishment of a back channel. Only proxy servers that maintain a table of expected return sockets can proxy services like H.323 (the protocol used by NetMeeting for voice and video conferencing).

Many services exist for which there are no effective content filters. Stream-based services like RealAudio or RealVideo are very difficult to filter for content because the content must stream through in real time, and an interruption in the compressed stream will make the remainder of the stream undecipherable. Since content like this cannot be reliably filtered, it should be blocked if considered a security threat.

### ***Proxies Do Not Protect the Base Operating System***

Proxy servers are based on web servers, and like web servers, they operate at the Application layer—above the Network and Transport layers. This means that they do nothing to filter TCP/IP packets that arrive at the server, and they don't interfere with other Application layer services like file sharing or remote procedure call interfaces.

This leaves the machine completely open to hacking, unless you take other measures to secure the machine. While most modern operating systems include support for packet filtering, their filters are usually not as robust as true firewalls. And you need to ensure that only those public ports that correspond to the services that you intend to proxy are open.

Some security experts recommend running the fewest possible number of services on a firewall, and separating proxy functionality on to separate machines under the presumption that filters should be as simple as possible to prevent their being exploited. The problem with this is that exploits can occur at any level, and if you put a proxy server behind a filter, the hacker is behind the filter if he exploits the proxy. By using firewalls with integrated proxy servers, the filters can still protect the network even if the proxy service is exploited.

### ***Lax Default Configurations***

Many proxy server software packages suffer from lax default configurations that can cause serious security problems. For example, WinGate, the most popular proxy server for home and small office environments, is used to share a single Internet connection rather than for security. For this reason, the software producer made it easy to set up for people who didn't understand proxies, and set it up to work by default for most common protocols.

For versions before 3.0, the default installation opened up a Winsock proxy to the external interface, which allowed hackers to connect to the external interface as if they were internal clients. The hackers could then use the proxy to connect to other web or Internet services as if they were working from the unsuspecting home user's computer directly. This effectively laundered their connection and made it appear as if the owner of the computer running WinGate were performing the illegal activities the hackers actually performed. Version 3.0's default configuration disabled connections coming from the external interface.

Many proxy servers suffer from the problem of lax default configuration because they are often designed for less experienced computer users and put performance and functionality ahead of security. Most can be configured correctly, but users frequently ignore the software once they've got it completely installed.

### **Performance Liabilities of Proxies**

Proxy servers only have one performance liability, but for the sake of uniformity, I'll present it as a bulleted list:

1. Proxy servers create a service bottleneck.

### ***Proxies Create a Network Bottleneck***

Like firewalls or routers, a single proxy server connection to the Internet can create a bottleneck if it's not properly upgraded as the number of network users' increases. Although proxies initially improve performance through their caching mechanism, you'll make everyone wait behind a slow

machine if you've got more clients than the server can effectively support. But beware of blaming your proxy for a bottleneck that's actually caused by a slow Internet pipe. If you have only one Internet connection, and it is a T1 (1.5MB) or slower connection, any computer that actually meets the minimum requirements for the operating system and the proxy server is fast enough to handle the load. Proxy bottlenecks only occur on network connections faster than 1.5MBs or when something is actually wrong with the proxy server.

This problem is easy to solve: add more proxy servers. Unlike websites or public servers, a proxy server doesn't need to have the exact same configuration across a number of machines. You can directly attach any number of proxy servers to your external network connection and assign each client inside your network to one of the servers on a random or fair share basis. For example, if you've got four proxy servers running, just assign every fourth client to the same proxy server. You'll lose some of the caching effect because a client on a different proxy who accesses a site won't make that site available to the other proxies.

You can also use sophisticated, high availability software and TCP/IP load balancing to handle the connection to multiple proxies, but that involves considerable expense and is not much more efficient. It does provide proxy redundancy though, because otherwise a segment of users would lose service if their assigned proxy went down.

### **Explicit vs. Transparent Proxies**

Most proxies, especially common HTTP proxies, require that the client software be explicitly configured to use the proxy server to access data (such as web pages) from outside the network. This means that not only must every web browser, FTP client, or videophone application you want to proxy have the ability to use a proxy server (many do not, having been naively programmed to expect unfettered access to the Internet), but also a system administrator must either configure all the applications on the client computers in the network to use the proxy or teach the users how to do it.

The configuration issue is such a burden to network administrators that modern web browsers have the ability to automatically detect proxy settings on a network. Other client software, such as FTP or Net2Phone, has not been reprogrammed to do this however. But while this feature is a boon to network administrators of web browsers, there is a better way that works for other protocols too and does not require configuring or modifying client network software—transparent proxies.

### ***Transparent Proxies Rewrite the Rules***

All modern firewalls can redirect incoming requests to certain ports to specific interior computers that will satisfy those requests (such as a web server on the interior network that is protected by the firewall). Similarly, a firewall can intercept and redirect outgoing traffic to a particular computer, such as a proxy server for web requests. The client computer need not know that its traffic has been intercepted because the firewall can redirect the proxy server's response back to the originating client as though nothing untoward had happened (using the same network address translation mechanisms that are now so wide-spread). There are instructions on the Internet for using the firewalling features of BSD or Linux along with separate proxy packages such as Jigsaw in this manner.

## **Proxy Best Practices**

Proxies are useful for a number of different purposes, and for that reason security often takes a backseat to performance or connection multiplexing. Proxies can be extremely dangerous if they're used incorrectly (okay, people can't actually get hurt just dangerous in the legal risk sense) because hackers can exploit them to make it appear as if their activities are coming from within your network. This can make your company liable for their activities.

### ***Use a Real Firewall***

The most important thing you can do to protect yourself is to either use the proxy functionality of a real firewall or put a firewall in front of your proxy server to protect it. There's no reason why a proxy server has to be directly connected to the external network unless the proxy is used for reverse proxy load balancing of a website.

### ***Disable Routing***

If you use proxies as your primary protection against hackers on the Internet, be sure you disable routing through the proxy. If you allow routing through the proxy, the proxy is not performing a significant security function for your network because your clients will all be directly addressable from the Internet. Proxies' client-hiding feature relies upon disabled routing to prevent a number of low-level protocol attacks.

Proxies are usually set up initially with routing disabled, but after some time a service or protocol might be needed for which you do not have a specific proxy service or which cannot be proxied. Don't be tempted to simply enable routing in this case. If you find you need services that cannot be proxied, use Network Address Translation. If the service can neither be translated nor proxied, don't use it at all.

### ***Secure the Base Operating System***

Securing the base operating system is crucial to the effective use of proxies as security devices. If hackers can exploit the server upon which your proxy runs, they can reconfigure the proxy security settings to bypass it completely.

This is especially important in Unix and Windows environments. Both operating systems are notoriously susceptible to well-known hacking exploits, so proxies that run upon them are just as susceptible.

Use strong user-based security permissions as well as port and protocol filtering at the operating system level to make sure your proxy server is serving only those protocols you intend for it to serve. Stay up to date on the latest hacking exploits for your operating system and be certain that you apply patches and hotfixes to your external security servers as they are released. It's more important for a publicly exposed server to be secure than it is to be stable. A crash due to an untested patch or hotfix only causes a temporary loss of service—it doesn't allow a security breach.

### ***Disable External Access***

Never allow external network clients to proxy through your server, even if it seems like it would make sense for remote users to do so. By allowing external proxy access to your server, you make it possible for hackers to exploit your proxy server to launder their IP connections and make it appear as if your proxy server is the origin of their attacks. This could make you legally liable for the damages they cause.

### ***Disable Excess Services***

Don't pile all your public services on the same machine as your proxy server. This general rule is especially important when applied to security mechanisms like proxy servers. If a service like FTP or SMTP allows a hacker access to your proxy server, the hacker can disable the proxy server's security settings to gain further access to your network. If these services are divided amongst several machines, however, an FTP-specific attack will only yield access to the FTP server not the rest of the network.

On Windows, it's especially important to unbind the NetBIOS session ports from the external TCP/IP interface through the network Control Panel. Leaving these ports open will make it possible for hackers to use automated password guessing tools to attempt to log directly into your proxy server. Once that's accomplished, they have free rein to modify your security settings.

~Article by Enuide written for Linux Exposed

This article comes from Linux Exposed

<http://www.linuxexposed.com>

The URL for this story is:

<http://www.linuxexposed.com/modules.php?op=modload&name=News&file=article&sid=551>