Global Information Assurance Certification Paper

Why Place Your Web Servers On the Web?

–A Look at Web Proxy Technology and Architecture

Written By: Darrin Mourer
Date: November 8, 2000

Introduction and Overview

Common to the headlines of today's news articles are those of web sites belonging to well-known businesses being brought down or defaced due to some sort of breach of security by a hacker. Even more alarming are the occasional reports of data or customer information being stolen during these attacks. This problem is a very real one for anyone tasked with ensuring the safety/uptime of a website. These attacks may be the result of operating system (OS) holes, web server software bugs, network design, etc. Each one presents a different challenge to overcome.

In the typical layout of the components of a website, you will have a perimeter defense mechanism such as a router and hopefully a firewall. Attached to this device you will have your web server farm with any application support servers (such as a database server) somewhere behind them. A hole is then opened on your firewall to allow http traffic (usually port 80 and 443) in to your web servers. Any data required not on the web server is retrieved from the supporting servers.

In the very short example laid out above there are several risks we can identify, even though they may be necessary for providing the information.

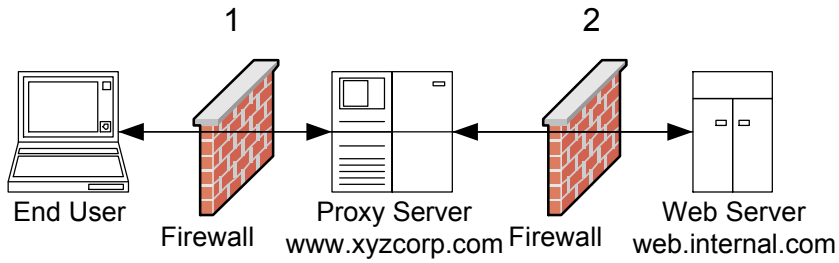
1) By opening a port on your firewall to these web servers, you are allowing direct access to your web servers for that service. This allows any hacker anywhere in the world to access your servers directly.
2) Your content is all located on the very same servers you are allowing access to. If any type of illegal access to this system occurs. The intruder has the ability to modify the actual contents of your site. Most commonly the primary page of a given site is vandalized to maximize exposure, which is contained on all your web servers for the given site.
3) Through simple reconnaissance, much can be obtained concerning your server and data layout.
4) The incoming requests for the data from your site are often not checked or scrutinized. Wise hackers employ ways of embedding different types of data into "normal" looking packets destined for their target. This altered data can contain information that the receiving server is not expecting, and may cause the server to react in an unforeseen way, such as revealing information about itself that it should not reveal, or allowing further access into the system.

Proxy Technology

By definition a proxy as pertains to data networks is – "A technique used to cache information on a Web server and acts as an intermediary between a Web client and that Web server. It basically holds the most commonly and recently used content from the World Wide Web for users in order to provide quicker access and to increase server security. They open a socket on the server, and allow communication via that socket to the Internet. For example, if your computer is inside a protected network, and you want to browse the Web, you would set up a proxy server on a firewall. The proxy server would be configured to allow requests from your computer, trying for port 80, to connect to its port 1080, and it would then redirect all requests to the proper places" [1]. Lets simplify this down into the important points. A proxy is a service handling the request made by another for a resource on a foreign client/host and serves as an intermediary between all systems involved.

The following diagram and explanation illustrate how the communications channel is handled and at what point certain tasks are performed.

Figure 1.0



In this example an end user is making a request for a document on the website www.xyzcorp.com. This request goes through the primary firewall to the proxy server. The proxy server at this point is acting on behalf of the "real" web server. The proxy server verifies it is a valid request by inspecting the different fields within the packet. If deemed a valid request the proxy will send a request to the internal web server web.internal.com for the same content. Once obtained, the proxy server will send the data back out as if it were the originating location.

Benefits

In the brief network layout explanation, four security risks were introduced. Each of the four individually is only a small threat. When you couple them together a better idea of how attackers are able to penetrate various sites on various platforms becomes very apparent. Each problem we have introduced will correlate which these groupings.

Direct access-

Web servers have typically been placed on the service network or "demilitarized zone" as some refer it. These severs are allowed direct access and usually have legal, routable IP addresses. Utilizing a proxy it is possible to remove these servers off of the public segment and place them back into your internal network (physical separation of the segment would always still be recommended). Once back into your internal network you are able to subject your web servers to very strict rules and policies, such as only allowing any type of network connection from the authorized proxy server. Your web servers will also no longer be subject to any types of scans or probes due to the fact that the proxy server will not pass these types of requests. The attacker would still be able to scan your proxy server, but this is of little concern, as it doesn't contain a full web server or any type of data.

Data placement-
Without a go-between on your site, data must be placed locally on the resource that is requesting it. If any type of unauthorized access is obtained on the server that is exposed, then that places anything on that server at risk. An even bigger problem is that once the primary server has been compromised, this places the other systems at risk. An attacker can make any type of request s/he pleases to a database server for example, and query it for information, or flat out mount/map a point on that server and copy part of the database. Many exploits exist that allow you to run arbitrary code on servers via "buffer overflows". This is simply adding in or submitting malformed requests to a program in which the program cannot handle properly due to a flaw in the programming. The accompanying code then has the ability to run as a higher privileged user or process. The code can contain instructions to perform a variety of tasks but normally they are used to subvert the OS into allowing further access to the attacker. Without the proxy in place the attacker has now gained the ability to run code on the system that contains your data.

Data layout-
The layout of your website ties in closely with the required placement of that data. Utilizing a proxy server in the middle, the true location of your data becomes hidden. The proxy server will not give away the true location of the information. It is sending back content to the end user from itself; it does not simply forward the request back. A user has no way of knowing that what they are seeing did not originate from the server they requested it from. Imagine a hacker trying to break in to a given site, they finally manage to find a way in and now have a look into the box they just broke into. They are greeted with no software that looks familiar, except maybe OS wise, no web server, and not a shred of useful data about anything other than the local machine and some network settings. As a side note, this scenario brings up another point. The attacker going after a website will be looking to perform web server based hacks and exploits. In their minds the server they are hitting is indeed the real thing and through some fingerprinting methods they have discovered it to be running Microsoft's IIS sever. They will then waste their time futilely running scripts, attacks, etc. based upon that information. Quite obviously these will all fail because the software does not even reside in that location. In this game, the less anyone knows about your network layout the better.

Exploits-
New exploits pertaining to the major operating systems and supporting applications are discovered every day. Computers directly exposed to the Internet are exposed to these attacks. Proxy technology will not completely solve this issue, as it still requires an underlying OS to run on. It is a much better scenario to have these attacks going toward/hitting a locked down proxy server as supposed to a web server. The proxy server will not be open to web server exploits, as it should only be running a single process. In addition, the attacks that it does receive for a web server that are disguised as normal traffic (which they often are) are not passed. The proxy will have a set of http instructions that are allowed and a definition as to what http should look like. A proxy will look at the packet and if it isn't a valid request, it is dropped. If the packet does appear valid but is malformed, the blemish is not regenerated when the request is made to the web server, eliminating any type of packet shaping attacks. A couple years back there was an attack introduced which came to be known as the Ping of Death. By resizing a single packet to exceed the maximum length and sending it in a ping request, you were able to crash the operating system. In our proxy layout, that packet, even formed as a web request, would never touch the web server. The proxy would simply make the proper request (if there was one included with the packet) and form its own carrier packet. Here the worst-case scenario becomes the ability to drop the proxy server with these attacks. This would be classified as a DoS or Denial of Service attack. While you may temporarily disrupt service, your data is still in a safe location and still available for when you are able to stop the attack.

Drawbacks-

There are a couple drawbacks to implementing this technology. Before implementing them into you environment you be aware of them.

1) Using proxy servers adds another layer of servers and software to your network. In addition these should always be put up in a redundant format to prevent a single point of failure. This can be a substantial cost and adds to administrative overhead.
2) Your users will be going through an additional layer, which has the potential to slow down response times to your data. Each proxy request must be made twice. You can always build up a bank of proxy servers to ensure this isn't an issue, but then cost increases. The ratio of proxy to web servers is low due to the fact that the proxy server does not have the overhead of running the web server processes or handling the data.

The Players-

In this arena there are two products that have implemented this technology.

Axent Webthority (formerly WebDefender)

Additional Benefits-
Centralized Management
Strong Authentication
Push User Content
Role Based Authentication
Multi-tiered Authentication
Replication and Load Balancing (intrinsic to product)
Works with any web server and requires no agent on each server

More information can be obtained from:
http://www.axent.com/Axent/Public/Main?nav=Products

Tivoli SecureWay Policy Director

Additional Benefits-
Centralized Management
Strong Authentication
Supports WAP
Can Authenticate Non-HTTP Sessions
Authentication API
Provides for Simple Access Control Policies
Can Create Logical Name Spacing

More information can be obtained from:
http://www.tivoli.com/products/index/secureway_policy_dir/index.html

Netegrity and Securant both have products in this space however both have not implemented proxy technology at this time.

Conclusion

Proxy technology while not new, is showing up in many facets of technology, proving it is contains additional benefits. A chief benefit for the years to come will be its inherent security features. It hides systems through obscurity, provides protection as a go-between, and serves to confuse by acting on behalf of other services. Four chief risks concerning websites can all be solved using this one technology. It provides a crucial extra layer of protection eliminating many common methods of attack. While this is an important step, there are many layers constituting a well-rounded security plan. Both of the products listed above are a great start. They are mature products that have been around a while. Look for this technology to make large inroads in the future where data security is of utmost importance. Following the conclusion is a short list of complimenting security products and how each might fit into the layered security model.

Security in Layers

"So what other technology can I implement that will compliment this investment?" is a common question. Looking back at figure 1.0 is a good starting point.

-Intrusion Detection Software (IDS)
Locating these attacks and being notified when they occur is essential. This allows you to take corrective action right then, whereas looking back later you may miss something. Attacks that any proxy might pass may be able to be detected using IDS software. Look for IDS software that can be configured to block that intruder at a perimeter firewall once an illegal scan/attack has occurred. Most can also close down the session immediately once something has been detected. Install the agents directly behind each firewall to verify you catch attacks coming from either direction. Host based IDS installed on each web/proxy server will allow you to take corrective action if a breach is ever detected.
-Perimeter
Ensure you are running an enterprise class firewall that has a long history behind it. The single biggest point to look for is that it scans up to and including the application layer. Also ensure your IDS can feed data into it.
-Assessment
Constantly scan your own system for possible holes. Several commercial scanners are available. Along with this make sure you have an adequate security policy in place to deal and react upon attacks

Resources

[1] Hunt, Craig. "Linux Network Servers." Apache Web Server. 24Seven. 1999.

[2] Unknown. "Tivoli SecureWay Policy Director Whitepaper." URL: http://www.tivoli.com/products/documents/whitepapers/sway_pol_dir_wp.html

[3] Unknown. "Tivoli SecureWay Policy Director Product Homepage." URL: http://www.tivoli.com/products/index/secureway_policy_dir/

[4] Unknown. "Axent Webthority Product Homepage." URL: http://www.axent.com/Axent/Public/Main?nav=Products

[5] Unknown. "Axent Webthority Technical Whitepaper." August, 2000. URL: http://www.axent.com/Axent/Public/Main?nav=Products&detail=/MainspanScripts/0/Axent/AxentWomSession/AxentWomSecureSession/N7OKLNFMHBA13C0N038BEH2643/WebthorityTechnicalSummary082200.doc

[6] , [1] Unknown. "Proxy Definition and Disclaimer." URL: http://www.proxy.net/proxy.html

[7] Mount Ararat Blossom. “NTBUGTRAQ Posting.” MS SQL Hacking. 14 Nov 2000. URL: http://www.ntbugtraq.com/default.asp?pid=36&sid=1&A2=ind0011&L=ntbugtraq&F=&S=&P=4815

[8] Mount Ararat Blossom. “NTBUGTRAQ Posting.” IIS Hacking. 18 Oct 2000. URL: http://www.ntbugtraq.com/default.asp?pid=36&sid=1&A2=ind0010&L=ntbugtraq&F=&S=&P=3993

[9] Unknown. “The World Wide Web Security FAQ.” URL: http://www.w3.org/Security/faq/wwwsf1.html#Q3

[10] Unknown. “Apache Web Server.” FAQ Rev 1.147. 12 Nov 2000. URL: http://httpd.apache.org/docs/misc/FAQ.html

[11] Simple Nomad. “Unofficial Web Hack FAQ.” Ver.3. 4. Mar 1998. URL: http://www.nmrc.org/faqs/www/index.html