Attackers And Their Tools: How Entercept Protects Servers

An
Entercept Security Technologies
White Paper

entercept
SECURITY TECHNOLOGIES

Computer attacks against servers account for billions of dollars of damage each year.
How do these attacks happen? What can businesses do to prevent them? This paper is intended to answer these questions by explaining the most common methods used to compromise servers and how Entercept prevents those attacks from succeeding.

The Attacker’s Toolbox

Many tools and attack methods are available to today’s attacker. The most common attacks targeting servers include:

- Worms
- Buffer Overflow Exploits
- Privilege Escalation Exploits
- Trojan Horses
- Rootkits
- Backdoors
- HTTP Exploits

Understanding each of these attack methods is essential to combating them.

Worms

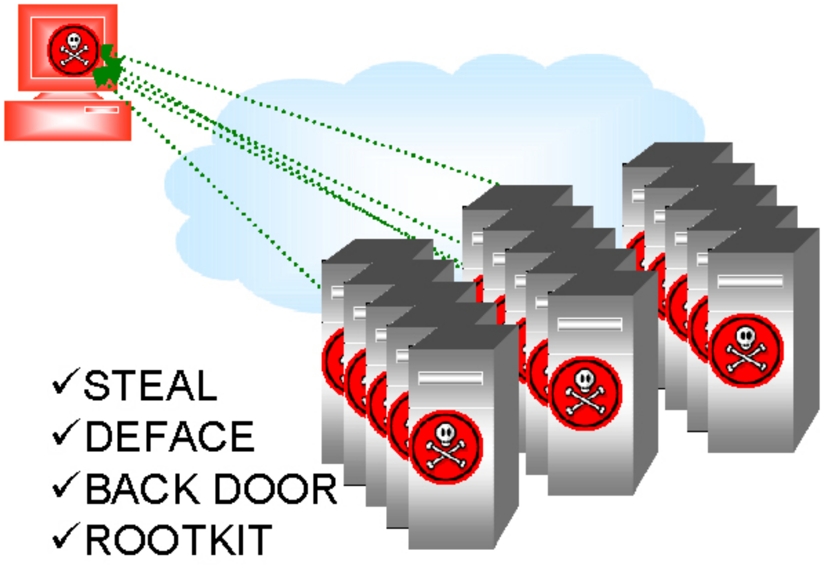
Worms are malicious programs that spread themselves automatically, as opposed to viruses, which are malicious programs that are spread by human intervention (inserting an infected floppy disk into a computer, double-clicking on an email attachment, etc.) Recent worms such as Code Red and Nimda have caused billions of dollars of damage, cleanup costs, and loss of business. Lately, attackers are using worms more frequently, since they can do so much damage so quickly.

Worms are very dangerous for several reasons. First, they spread very quickly. Code Red infected over 100,000 machines in 24 hours. Second, they can generally perform any malicious activity the attacker desires if the worm is able to gain sufficient privileges. Third, they are becoming easier to develop, with worm-generating programs known to be circulating on the Internet.

A worm has three main parts:

- Enabling Vulnerability – The "hole" that the worm exploits in order to gain access to the system
- Spreading Mechanism – The method by which the worm chooses and communicates with its victims
- Malicious Payload – The actual damage that the worm does once it compromises a system.

These three parts differ from worm to worm, but all worms have these three elements.



Buffer Overflow Exploits

Buffer overflow exploits are one of the largest problems in computer security today. In all application programs, there are buffers that hold data. These buffers have a fixed size. If an attacker sends too much data into one of these buffers, the buffer overflows. The server then executes the data that ‘overflowed’ as a program. This program may do any number of things, from sending passwords to Russia to altering system files, installing backdoors, etc., depending on what data the attacker sent to the buffer.

Programmers can prevent buffers overflows by checking the length of the data submitted to the buffer before storing it in the buffer. If the data is too large, it returns an error. Unfortunately, many programmers forget to check the length of the data before saving it to a buffer. Thus, applications contain a large number of "unchecked buffers," which are vulnerable to attack. Microsoft has released at least five bulletins in the past six months regarding unchecked buffers that exist in their products. When a vendor (Microsoft or any other vendor) releases a patch to stop these potential buffer overflows, the patch simply adds code that checks the length of the data before it saves it to the buffer. Thus, if a patch is available, a patch will prevent a buffer from being overflowed.

Buffer overflow exploits are such a large problem for several reasons:

- *Buffer overflow exploits are very common.* There are hundreds of known unchecked buffers that can be overflowed by hackers with more being discovered all the time. Over 50% of the CERT advisories deal with buffer overflow exploits.

- *Buffer overflow exploits are easy to use.* Anyone (10-year olds and script kiddies included) can download buffer overflow attack code and follow a simple "recipe" to execute it. No advanced technical knowledge is necessary to run pre-written buffer overflow exploit programs.

- *Buffer overflow exploits are very powerful.* In many cases, the malicious code that executes as a result of a buffer overflow will run with administrator-level privileges, and therefore can do anything it wants to the server.

> “[In 2001] there was a 33 percent increase in the number of organizations hit with buffer-overflow attacks…”
>
> ***2001 Information Security Magazine Industry Survey***

Privilege Escalation Exploits

Privilege escalation exploits grant administrator or root-level access to users who previously did not have such access. For example, an account exists on all Windows NT and 2000 servers called “Guest”. This account, by default, has no password. Anyone can log on to the server using this “Guest” account and then use a common privilege escalation exploit called “GetAdmin” to gain administrator-level access to the system. Many other privilege escalation exploits exist, such as HackDLL and others. These exploits are very useful, since they allow anyone who has any level of access to a system to easily elevate their privilege level and perform any activities they desire.

Trojan Horses

In the oft-repeated story of the Trojan horse, invaders used something that was seemingly benign (a large wooden horse) as a vehicle to attack a fortified city. Similarly, the Trojan horses of the information security world are seemingly benign programs that attack computer systems.

Commonly, computer attackers replace key system files and/or programs with malicious versions. When these programs are executed, they perform their predetermined destructive activities, and users are powerless to stop them.

For example, an attacker could replace one of the Windows operating system DLL’s (Dynamically Linked Library) with a malicious version. DLL’s are program files that Windows calls on to perform various tasks. An attacker may replace one of these DLL’s with a Trojan horse version that does everything the normal DLL did, and a little more. That little more may be any number of things, from reformatting the hard drive to stealing credit card numbers, etc.

Backdoors

When attackers obtain root-level access to a server (using a buffer overflow exploit or a privilege escalation exploit, for example) they will want to do two things:

1 – Install a backdoor
2 – Cover their tracks

Backdoors allow attackers to remotely access a system again in the future. For example, the attacker may have used a particular security hole to get root-level access to a computer. However, over time, that particular security hole may be closed, preventing the attacker from accessing the system again. In order to avoid being shut out in the future, attackers install backdoors. These backdoors take different forms, but all allow an attacker to access the server again without going through the standard login procedures and without having to repeat the attack that gave them access in the first place.

Many worms install backdoors as a part of their malicious payload. Code Red II, for example, installed a backdoor that provided access to the C and D drives of the compromised Web server from anywhere on the Internet. Other common backdoor programs are Netbus and BackOrifice, which allow attackers to remotely control a compromised server.

Rootkits

Rootkits are used to cover an attacker's tracks. If an attacker installs a backdoor or other malicious program, the system administrator may notice the new program and remove it, ending the hacker's ability to access the system in the future. The goal of a rootkit is to disguise the existence of malicious programs on a system.

By replacing certain system programs with modified versions of those same programs, rootkits mask the presence of backdoors or other malicious programs. For example, the UNIX program "ls" prints a directory listing of the file system. This would normally allow a sysadmin to see files left by an attacker. The rootkit installs a modified version of "ls" that displays all the files and programs in the directory *except* the backdoor program and any other files left by the attacker. This effectively masks the evidence of the system compromise. Rootkits generally replace "ls" as well as many other operating system programs to cover their tracks.

HTTP Exploits

HTTP exploits involve using the Web server application to perform malicious activities. These attacks are very common and are growing in popularity because firewalls typically block most traffic from the Internet to keep it away from corporate servers. However, HTTP traffic, used for Web browsing, is almost always allowed to pass through firewalls unhindered. Thus, attackers have a direct line to the Web server. If they can coerce the Web server into performing malicious activities, they can access resources that would otherwise be unavailable.

New HTTP exploits appear quite frequently. Some recent exploits include the Unicode Directory Traversal Exploit and the Double Hex Encoding Exploit. Directory traversal exploits use strings like '../../../' to access directories outside the normal webroot directory where Web content is stored. Since most Web servers will block URLs that contain '../,' attackers circumvent this protection by using Unicode or hexadecimal encodings to represent the '../' pattern. By typing a properly crafted attack string into a Web browser, attackers can access other directories on the Web server. These other directories may contain confidential information, passwords or other sensitive files. By using an HTTP exploit, attackers can access these files easily through a standard Web browser. Other HTTP exploits allow attackers to execute programs, alter system information, access registry keys, and perform other malicious activities.

Entercept Protects Your Servers

Entercept protects servers from the types of attacks mentioned above as well as many others, including new attacks that have yet to be discovered. Examining Entercept's architecture and multiple layers of protection reveals how the product blocks these attacks.

Entercept resides adjacent to the operating system and intercepts system calls prior to their execution. If the call is determined to be an attack, Entercept blocks the call; otherwise, it permits the call to proceed normally.

Entercept is available in two versions: the Standard Edition and the Web Server Edition. The Web Server Edition includes all the functionality of the Standard Edition as well as additional features specific to preventing attacks against Web Servers.

Entercept Standard Edition

The Entercept Standard Edition protects the most important part of any server: the operating system. All users and programs access the server through the operating system.

Resource Protection

The Standard Edition protects system resources (libraries, files, directories, user accounts) and prevents them from being altered. This protection is extremely valuable, since Trojan horses, rootkits, and backdoors alter the system resources in order to install themselves. By preventing alteration of these resources, the Entercept Standard Edition prevents the installation of these pervasive hacking tools.

Stopping Privilege Escalation Exploits

The Standard Edition also prevents privilege escalation attacks from succeeding. Privilege escalation attacks are very common, since they give ordinary users superuser-level (root or administrator) access to the server. The Entercept Standard Edition prevents these attacks from succeeding by preventing access to the files and resources necessary to alter privilege levels. Even new, previously unpublished privilege escalations can be stopped without knowledge of the specific exploit. This is possible since all privilege escalation exploits alter user privileges, and Entercept prevents such alterations.

Buffer Overflow Exploit Prevention

Buffer overflow exploits are the most common method of attacking servers today. These attacks can be downloaded and executed easily by very unsophisticated attackers, sometimes called “Script Kiddies.” Over 60% of the CERT advisories deal with buffer overflow exploits, so preventing these very common exploits is critical. The Entercept Standard Edition is able to determine if code that is about to be executed by the OS came from a normal application or from an overflowed buffer. If the code came from a normal application, Entercept allows it to be executed. If it came from an overflowed buffer, it is blocked, and the buffer overflow exploit is thwarted. Thus, Entercept protects the server from being compromised as a result of the overflowed buffer. This protection is extremely important, since it stops the most common method of attacking servers.

Unknown Attacks

Most importantly, Entercept can prevent the aforementioned attacks using behavioral rules technology, rather than relying solely on individual signatures. This technology allows Entercept to stop new and previously unknown attacks without requiring signature updates to the product. For example, Entercept’s rules to stop buffer overflow exploits from succeeding are not tied to a specific application or signature. Instead, Entercept can prevent buffer overflow exploits from succeeding, regardless of the application or buffer involved. Similarly, Entercept’s resource protection protects against new attacks as well as against older, known attacks.

Entercept Web Server Edition

The Entercept Web Server Edition (WSE) runs on Solaris 2.6, 7, and 8 as well as Windows NT/2000. It supports Netscape Enterprise Server, iPlanet Web Server, Apache Web Server, and Microsoft's IIS Web Server. The Entercept Web Server Edition (WSE) layers are:

HTTP Filtering

Entercept Web Server Edition includes an HTTP filtering layer that intercepts HTTP requests after they are decrypted and decoded (from any SSL encryption, Unicode encoding, or hex encoding) but before the Web server executes them. Entercept uses signatures at this layer to detect attacks against the Web server and other vulnerabilities. The value of this filtering was proven during the recent Code Red and Nimda worm attacks. Entercept blocked both worms at the HTTP layer, before anyone knew anything about them. No signature updates were required, since Entercept's HTTP filtering protects against the abnormal requests that these worms used to attempt to penetrate the Web server. Neither Code Red nor Nimda infected servers that were protected by the Entercept Web Server Edition. This layer is the preferred place to stop attacks, since they are blocked long before the server can actually execute them.

Web Server Shielding

Entercept also employs Web Server Shielding to stop both known and unknown attacks from altering Web content or using the Web server as an attack tool. Entercept places the Web server application, its files, and its resources inside a conceptual steel vault. If the Web server attempts to access any resources outside that vault, Entercept blocks the attempt. Conversely, if any other user or process tries to access or alter the files or resources contained within the vault, Entercept blocks that access as well.

Entercept accomplishes this protection by defining a set of behavioral rules for the Web server. If the Web server attempts to do things that are not within its defined behavior, the attempt is blocked. This allows Entercept to protect against unknown, yet-to-be-discovered attacks. Instead of focusing on solely signature-based approaches like traditional IDS vendors, Entercept uses behavioral rules to define known, appropriate behavior. When a new attack is invented, it will, by definition, violate the Entercept rules that define appropriate behavior and will be blocked.

Signature-based approaches focus on *how* an attack works, trying to detect certain strings or other identifying information. This approach works, to some extent, for known attacks. However, if the attacker makes even minor changes to how the attack works, the previously written signatures no longer detect the attack.

Entercept focuses instead on *what* an attack does. Even if an attacker changes the *how* of an attack, the *what* remains the same: the attack does something malicious. Entercept's behavioral rules technology identifies attempts to perform malicious activities and prevents them from succeeding. This approach protects users against unknown attacks that are yet to be discovered.

Includes Entercept Standard Edition Protection Layers

The WSE includes all the functionality of the Standard Edition as well as additional levels of protection specifically tailored for Web servers.

Summary: How Entercept Blocks the Bad Guy Tools

- **Worms** – Entercept stops a worm from infecting a server by blocking the worm's attempt to exploit vulnerabilities on the server.

- **Buffer Overflow Exploits** – Entercept prevents execution of code as a result of buffer overflows, preventing compromise of the server.

- **Privilege Escalation Exploits** – Entercept prevents privilege escalation exploits via its Resource Protection layer that prevents alteration of system resources.

- **Trojan Horses, Rootkits, and Backdoors** – Entercept prevents these very common attack tools from compromising servers by preventing change to system resources. Since Trojan horses, rootkits and backdoors all attempt to modify system resources, Entercept prevents them from being installed.

- **HTTP Exploits** – Entercept's HTTP Filtering layer prevents HTTP exploits from succeeding by blocking malicious HTTP requests.

Entercept Security Technologies
www.entercept.com

US and Canada	ph: 1.800.599.3200	fx: 1.800.308.3777	email: sales@entercept.com
Europe	ph: + 44 208 387 5502	fx: + 44 208 387 5505	email: sales@entercept.co.uk