CMSC 414 — Computer and Network Security

Viruses and Other Malware

Dr. Michael Marsh

September 11, 2017

Malware

Malware is any **mal**icious soft**ware** that runs on a victim's system.

General categories:

Virus hides in stored code, propagates based on user action

Worm hides in running code, propagates automatically

Trojan immitates legitimate SW, typically propagated by attacker

Some malware combines these characteristics

Worms

Hide on a system, running behind the scenes

Try to replicate themselves onto other systems

May be a stand-alone program:
`/etc/rc3.d/S00malservd`

Usually try to hide themselves from `ls`, `ps`, `netstat`, ...

Attempt to *spread rapidly*

Trojans

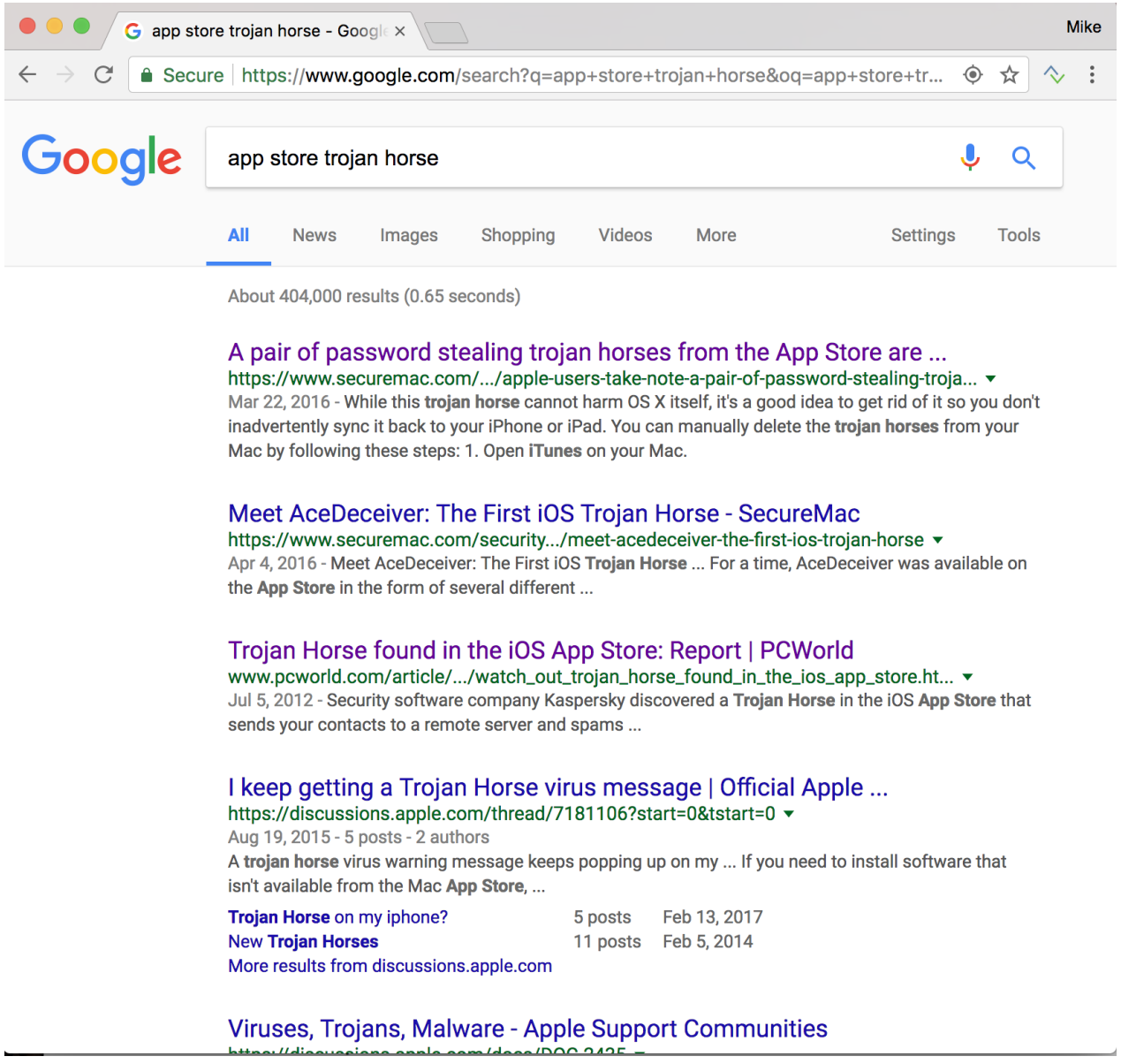
Run by user, but do more than advertised

```
void game() {
    struct Move* m;
    while( m = getMove() ) {
        stealUserInfo();
        processMove(m);
    }
}
```

May look similar to "real" programs

- "Fileless Trojan Kovter Poses as Firefox Update," *SecurityWeek*, July 2016
- "Mozilla Firefox for Mac free download. Always available from the Softonic servers"
- "Angry Birds Transformers Trojan targets Android, warns 'Obey or be hacked'," *Computerworld*, October 2014

Trojans


app store trojan horse - Google
Mike
Secure https://www.google.com/search?q=app+store+trojan+horse&oq=app+store+tr...
Google
app store trojan horse
All News Images Shopping Videos More Settings Tools
About 404,000 results (0.65 seconds)
A pair of password stealing trojan horses from the App Store are ...
https://www.securemac.com/.../apple-users-take-note-a-pair-of-password-stealing-troja...
Mar 22, 2016 - While this trojan horse cannot harm OS X itself, it's a good idea to get rid of it so you don't inadvertently sync it back to your iPhone or iPad. You can manually delete the trojan horses from your Mac by following these steps: 1. Open iTunes on your Mac.
Meet AceDeceiver: The First iOS Trojan Horse - SecureMac
https://www.securemac.com/security.../meet-acedeceiver-the-first-ios-trojan-horse
Apr 4, 2016 - Meet AceDeceiver: The First iOS Trojan Horse ... For a time, AceDeceiver was available on the App Store in the form of several different ...
Trojan Horse found in the iOS App Store: Report | PCWorld
www.pcworld.com/article/.../watch_out_trojan_horse_found_in_the_ios_app_store.ht...
Jul 5, 2012 - Security software company Kaspersky discovered a Trojan Horse in the iOS App Store that sends your contacts to a remote server and spams ...
I keep getting a Trojan Horse virus message | Official Apple ...
https://discussions.apple.com/thread/7181106?start=0&tstart=0
Aug 19, 2015 - 5 posts - 2 authors
A trojan horse virus warning message keeps popping up on my ... If you need to install software that isn't available from the Mac App Store, ...
Trojan Horse on my iphone? 5 posts Feb 13, 2017
New Trojan Horses 11 posts Feb 5, 2014
More results from discussions.apple.com
Viruses, Trojans, Malware - Apple Support Communities

Viruses

"Infect" executable files (or parts of files):

- Binaries
- Macros, such as in Word or Excel documents
- Javascript, such as in PDF files
- Boot sector, before the OS loads

```
void func(int b) {
    int a = 0;
    if ( (a-b) > 3 ) {
        printf("some condition\n");
        push %eip
        jmp 0xbaddbadd
        g();
    }
}
```

Attempt to *evade detection*

Virus Features

We will look at two *orthogonal* aspects of viruses:

- How do they propagate?
- What else do they do?

Infection Vectors

May be used by viruses, worms, or trojans

Vulnerable Network Services Frequently used by worms, often buffer overflows are involved

Backdoor Logins Malicious developers leave themselves a way to get in, whether on systems where they worked or through **code they've distributed**; sometimes this is laziness, rather than malice (eg, **default or unchangeable passwords**)

Social Engineering **Phishing**, copycat websites, **bad attachments**

Trojan Horses Once installed, can continue to infect

Physical Access **USB thumbdrives**, unsecured terminals, network ports, other peripherals

Malware Effects

Malware authors have many different goals:

- System disruption, such as annoying pop-ups
- Defacement of a publicly visible service
- Destruction of data
- Crashing a system
- Stealing data ("exfiltration")
- Hidden malicious services to send spam, etc.
- Ransomware to extort money in exchange for decrypting data
- Rootkits for later access or man-in-the-middle attacks

Malware Triggers

Malware may be triggered:

Immediately once installed

With a **Time-Bomb** that waits for a certain time or duration before acting

With a **Logic-Bomb** that waits for a certain set of conditions to be true

Manually through a backdoor or other mechanism the attacker installed as part of the malware

Group Exercise 1

Clone `metamorphic` using `get-assignment`.

Task 1 involves comparing compiled C code with assembly. Make sure you understand what you are seeing, and discuss within your group as necessary.

Malware Case Studies

Let's look at some historical instances of malware, some old, some recent.

Some of these are still found in the wild after a decade or more.

Morris Worm

We briefly talked about this before Labor Day

Written by Robert Morris in 1988 while a grad student at Cornell

Intended to measure the size of the Internet

Asked exploited system if it was infected

"yes" $\Rightarrow$ still adds itself with probability 1/7

Unintended denial-of-service from multiple copies running

Considered first wide-spread malware

1260 Virus

Written by Mark Washburn in 1989

First virus to use polymorphic encryption

Created for demonstration purposes as a proof-of-concept

ILOVEYOU Worm

Originated in the Philippines in 2000

Visual Basic script in an email attachement named "LOVE-LETTER-FOR-YOU.txt.vbs"

Opening the attachment on a Windows machine causes it to email itself to victim's entire address book with subject line "ILOVEYOU"

Massive social-engineering attack — spread faster than any previous worm

Similar worm in 2001 called "Anna Kournikova"

Code Red Worm

Buffer overflow attack against Microsoft IIS in 2001

Defaces infected website with

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

Spent about 3 weeks/month trying to spread

Spent about 1 week/month launching DoS against several IP addresses

Did not attempt to avoid non-vulnerable servers (like Apache httpd)

Mydoom Worm

Email spam worm released in 2004

Masquerades as email server errors, with subject lines like:

- "Error"
- "Mail Delivery System"
- "Test"
- "Mail Transaction Failed"

Sends itself (and malicious attachment) to victim's address book

Also spreads via KaZaA P2P network

Payloads include a backdoor and a DoS attack

Storm Worm

Appeared in 2007

Another social engineering attack using email attachments

Initial version purported to contain a story about a massive storm that had impacted parts of Europe

Subsequent versions varied the subject

Notable for adding victims' computers to a massive **botnet**, where the computers became **zombies** to be used in DoS attacks

Koobface (2008/2009) is similar, but spreads through Facebook messages

Zeus Trojan

Appeared in 2007

Browser keystroke logging and form grabbing

Mainly used to steal banking credentials

Uses **drive-by downloading** (unintended downloads) and **phishing** to spread

Conficker Worm

Appeared in 2008

Dictionary attacks against Windows admin passwords

Adds victims to a botnet

Many different generations, varying attack and *defense* methodologies

Stuxnet Worm

Appeared in 2010

Targets programmable logic controllers

Four different 0-day exploits

Payload geared to centrifuges — alters rotor speed in order to induce destructive vibrations

Payload only executed on Iranian centrifuges

Believed to be joint US-Israeli government operation

Flame worm (2012) believed to be closely related

Mirai

Appeared in 2016

Targets older versions of Linux, commonly used in Internet-of-Things (IoT) devices

Created massive botnet — most massive Distributed Denial-of-Service (DDoS) attacks ever seen

Memory-resident, so rebooting ejects it, but reinfection likely

IoT devices generally cannot be updated or secured

WannaCry Worm

Appeared in 2017

Ransomware — encrypts victim's files

Demands payments in Bitcoin in exchange for the decryption key

Ransomware has historically had a spotty record of working decryption keys

Infected medical, Telecom, and other large-scale businesses

Contained a killswitch — a domain that did not exist, but was later registered by a security researcher

Mirai botnet has attacked this domain

There is a simple thing you can do to never have to worry about ransomware: **back up your data!**

Group Exercise 2

Task 2 involves taking the assembly code you were given, and producing new generations of the code that does the same thing, but that looks potentially very different. Work with your group to come up with as many versions of the program as you can. Don't forget to commit your changes, so that you don't lose them!

Malware Detection

Malware initially detected based on *signatures*

- Byte sequences
- Encrypted byte sequences
- Code patterns
- Size differences before/after infection

Code Red Signature

Fairly typical buffer overflow:

```
GET /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN%u9090
%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090
%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u0003
%u8b00%u531b%u53ff%u0078%u0000%u00=a HTTP/1.0
```

No changes until Code Red II $\Rightarrow$ easy to detect!

Malware Detection

Later, *heuristics* were added

- Suspicious API call patterns
- Unusual OPCODE sequences
- Suspicious call flows
- Coincidence with known malware

Malware Detection

Many anti-virus products now *crowd-source* detection

- New patterns detected by software on customers' machines
- Upload to vendor's servers
- Updates pushed to customers to provide rapid protection
- Masquerade as DNS requests to avoid filtering

This sometimes finds and protects against new viruses before human detection

DNS Tunneling Example

Bogus hostname

```
a-0.19-b3000081.a010083.15e0.1d99.36d4.210.0.ic7arfsqqzf694fs8zf8nz2t9b.avts.mcafee.com
```

McAfee's DNS server for `avts.mcafee.com` interprets this as data

Protecting against Malware

Detection is an important component

Defense in Depth — Don't rely on one mechanism, create layers of defense, for when one of them fails

OS-level protection mechanisms:

- Jails
- Virtual Machines
- Containers

Jails

Most POSIX systems have `chroot` command

Creates a restricted filesystem, with a new root (/) directory

Only files in `chroot` **jail** accessible to user

- regular files
- directories
- devices (including network)

Allows for (limited) **privilege separation**

Group Exercise 3

Clone `chroot` using `get-assignment`.

Task 1 is to create and explore a chroot jail. The included scripts will help you get started. Explore the filesystem and general environment that you see in the chroot jail.

Jail Limitations

Still running on “bare metal”

Access to the filesystems/devices/CPUs/RAM of host

Tends to be obvious under inspection

Sometimes good enough

- simple malware
- benign programming errors
- inattentive attackers

Virtual Machines

Allows you to run a *guest* operating system on a *host* machine

Software on host is called a **hypervisor** (because it's like a supervisor, but more so)

Completely virtualized hardware world, including the CPU and RAM

Guests are isolated from each other, and from the host — much more complete separation than a chroot jail

Hypervisor controls all access to actual hardware

No special relation between host/guest OSes

Virtual Machine Limitations

The more you virtualize $\Rightarrow$ more resource consumption

Trade off efficiency with detectability

Newer CPUs provide hardware support $\Rightarrow$ closer to full virtualization without all of the penalties

Can also virtualize only certain types of resources

Containers

Fairly recent development

Requires special kernel features — newer Linux kernels (past several years)

Uses chroot, userspace filesystems, process groups to isolate container processes

Container OS must (for Linux containers) be Linux

CPU and RAM not virtualized

Start up very quickly, low overhead

Less isolation than VMs, but more than chroot

Container Limitations

Host/container OS must be the same (within limits)

Container's processes all visible to host

Network stack (potentially) visible to containers

Memory protection essentially the same as for chroot jails

Breaking Out

None of these isolation techniques is perfect

Clever malware will attempt to determine that it's in an isolation environment, and look for holes through which to escape

- syscalls
- shared folders
- memory attacks
- guest/host interaction mechanisms

CVE-2008-0928 Qemu 0.9.1 and earlier does not perform range checks for block device read or write requests, which allows guest host users with root privileges to access arbitrary memory and escape the virtual machine.

CVE-2015-3629 Libcontainer 1.6.0, as used in Docker Engine, allows local users to escape containerization ("mount namespace breakout") and write to arbitrary file on the host system via a symlink attack in an image when respawning a container.

CVE-2016-9962 RunC allowed additional container processes via 'runc exec' to be ptraced by the pid 1 of the container. This allows the main processes of the container, if running as root, to gain access to file-descriptors of these new processes during the initialization and can lead to container escapes or modification of runC state before the process is fully placed inside the container.

Group Exercise 4

Task 2 is to break out of your chroot jail, using tools that were left in place for you.

Breaking In

Generally easier than breaking out

Intention is ordinarily to protect the host from malicious guests/containers/processes

Jails provide no protection

Container systems might give ordinary users unlimited access to containers, root generally always unlimited

Virtual machines provide some protection, but root compromise on host generally means access to guest

The Pivot

Pivoting refers to using access to one resource as a way to gain access to additional resources

Can be machine-to-machine

Can also be guest-to-host-to-guest

Breaking out of a jail/VM/container is a pivot

Breaking into a VM/container is another pivot

Frequently, pivoting from *dual-homed* systems into an *internal network*

More on this when we get to network attacks

Group Exercise 5

Task 3 is to modify you chroot jail in order to make escape harder. Within your group, discuss various ways of breaking out of the jail, and how you might prevent these from being accessible.