

Lecture 11 – Malware

Stephen Checkoway

University of Illinois at Chicago

CS 487 – Fall 2017

Slides adapted from Michael Bailey

Malware definition and goals

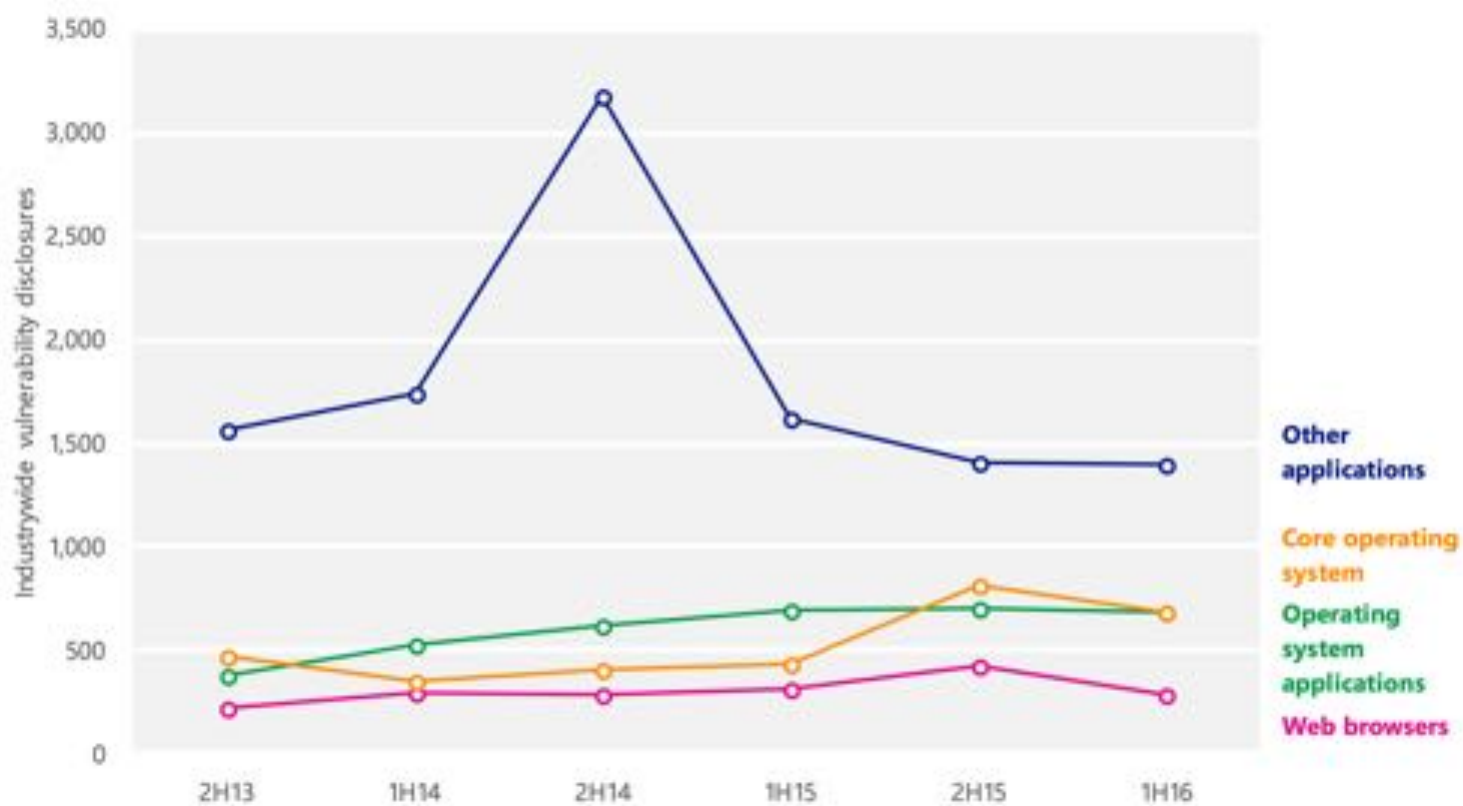
- What is malware?
 - Set of instructions that run on your computer and do something an attacker wants it to do.
- Muddled Taxonomy, but difference primarily
 - How they get on your machine
 - What do they do

Encounter rate trends for the locations with the most (Windows) computers reporting malicious and unwanted software encounters, by number of computers reporting

Country/Region

Country/Region	3Q15	4Q15	1Q16	2Q16
United States	10.8%	12.5%	11.9%	12.0%
China	14.9%	18.9%	19.1%	21.1%
Brazil	29.2%	34.4%	29.9%	29.4%
Russia	22.8%	28.7%	27.2%	24.9%
India	36.5%	44.2%	35.4%	32.6%
Turkey	32.6%	40.3%	34.8%	31.4%
France	18.8%	19.4%	17.0%	15.3%
Mexico	23.9%	28.5%	24.4%	23.8%
United Kingdom	11.9%	13.9%	13.7%	11.5%
Germany	12.2%	13.8%	13.0%	13.0%
Worldwide	17.8%	20.8%	18.3%	21.2%

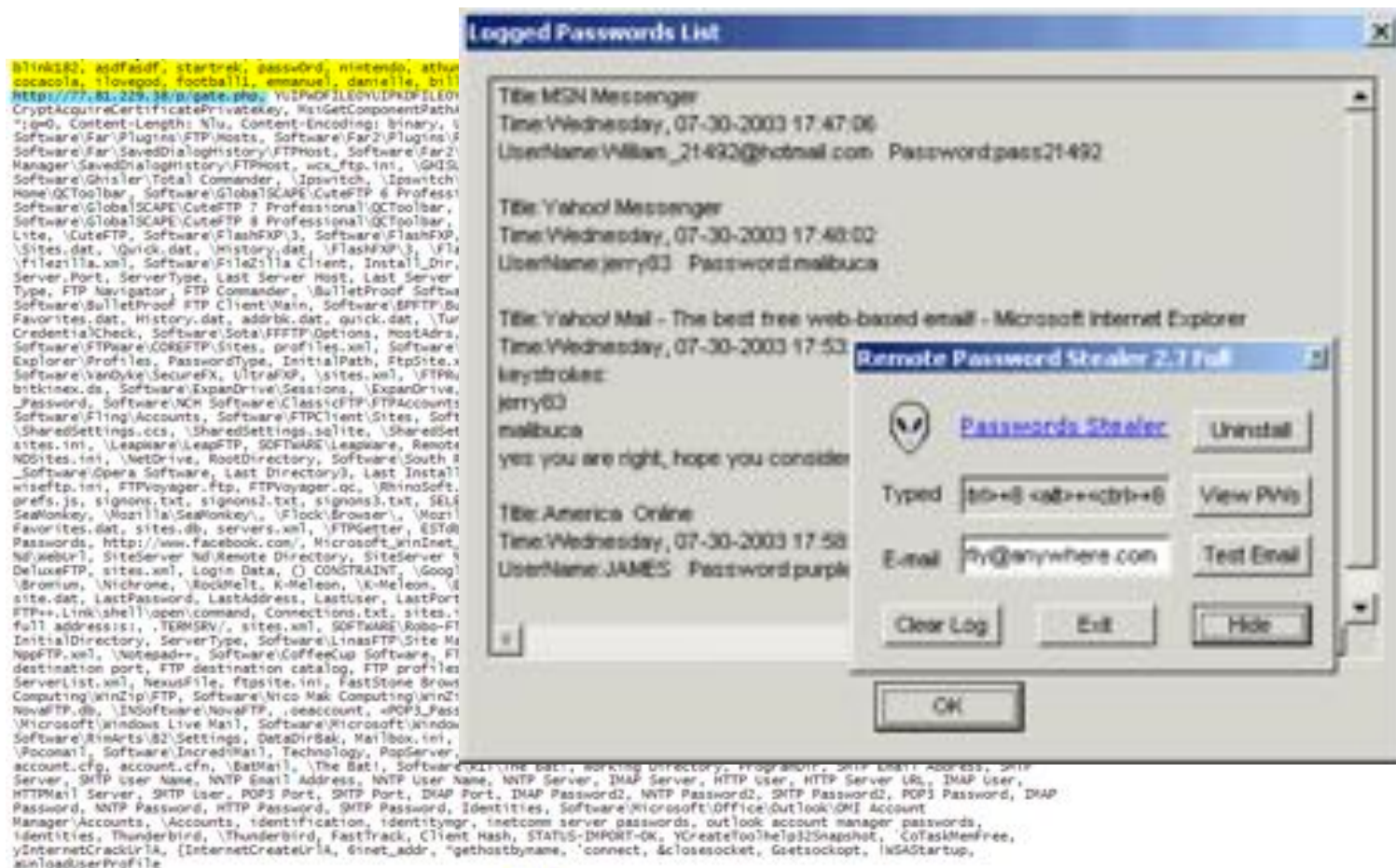
Industry-wide operating system, browser, and application vulnerabilities, 2H13–1H16



What Can Malware Do?

- Pretty much *anything*
 - Payload generally decoupled from how manages to run
 - Only subject to permissions under which it runs
- Examples:
 - Brag or exhort or extort (pop up a message/display)
 - Trash files (just to be nasty)
 - Damage hardware (Stuxnet)
 - Launch external activity (spam, *click fraud*, DoS)
 - Steal information (*exfiltrate*)
 - Keylogging; screen / audio / camera capture
 - ***Robbins v. Lower Merion School District***
 - Encrypt files (*ransomware*)
- Possibly delayed until condition occurs
 - “time bomb” / “logic bomb”

Key logging and Password Stealing



Logic Bombs

- A **logic bomb** is a program that performs a malicious action as a result of a certain logic condition.
- The classic example of a logic bomb is a programmer coding up the software for the payroll system who puts in code that makes the program crash should it ever process two consecutive payrolls without paying him.
- Another classic example combines a logic bomb with a backdoor, where a programmer puts in a logic bomb that will crash the program on a certain date.



The Omega Engineering Logic Bomb



- An example of a logic bomb that was actually triggered and caused damage is one that programmer Tim Lloyd was convicted of using on his former employer, Omega Engineering Corporation.
- On July 31, 1996, a logic bomb was triggered on the server for Omega Engineering's manufacturing operations, which ultimately cost the company millions of dollars in damages and led to it laying off many of its employees.

A view into a network attack

Net administrator charged in \$10M "logic bomb" case.

By Ellen Messner
Bridgeport, Conn.

In one of the costliest reported acts of computer sabotage, an engineering company next month will prosecute its former network administrator for electronically destroying computer files that the company claims cost it about \$10 million in sales.

Omega Engineering, Inc., is set to go to trial against Timothy Lloyd, the chief network program designer, who the company said planned a LAN-based logic bomb that went off after his job was terminated. The logic bomb wiped

out all the files on the company's Novell, Inc., network-based servers.

What detonated the Omega bomb was not immediately clear.

Security experts said a logic bomb usually is a software program that, once activated by a specific date for example, erases through files or reformat hard drives. The bomber's intent is to hopelessly damage and erase data.

"[Logic bombs] can be as simple as a script that runs a bunch of delete commands," said Chris Barnes, vice president for servers and system management strategy.

See [Results](#), page 16.



The Omega Bomb Code

- The Logic Behind the Omega Engineering Time Bomb included the following strings:
 - 7/30/96
 - Event that triggered the bomb
 - F:
 - Focused attention to volume F, which had critical files
 - F:\LOGIN\LOGIN 12345
 - Login a fictitious user, 12345 (the back door)
 - CD \PUBLIC
 - Moves to the public folder of programs
 - FIX.EXE /Y F:*.*
 - Run a program, called FIX, which actually deletes everything
 - PURGE F:\ALL
 - Prevent recovery of the deleted files

LOGIC BOMB SET OFF SOUTH KOREA CYBERATTACK



A disconnected computer monitor is seen at a newsroom of Korean Broadcasting System (KBS) at its headquarter in Seoul, South Korea, Wednesday, March 20, 2013. Computers networks at two major South Korean banks and three top TV broadcasters went into shutdown mode en masse Wednesday, paralyzing bank machines across the country. Photo: AP/Kim Ju-sung, Yonhap

Ransomware



Petya Ransomware

The harddisks of your computer have been encrypted with an military grade encryption algorithm. There is no way to restore your data without a special key. You can purchase this key on the darknet page shown in step 2.

To purchase your key and restore your data, please follow these three easy steps:

1. Download the Tor Browser at "<https://www.torproject.org/>". If you need help, please google for "access onion page".
2. Visit one of the following pages with the Tor Browser:

<http://petya37h5tbhgyki.onion/P9UUR3>

<http://petya5koahsf7sv.onion/P9UUR3>

3. Enter your personal decryption code there:

cdSPP4-JUZrRr-pMSxia-gXpmfB-uGWorF-FfMphi-XTUzVn-QwFeeV-ofb94y-HuScaa-rBignU-djYAEH-BWEakz-wrQ85W-BbsCzu

If you already purchased your key, please enter it below.

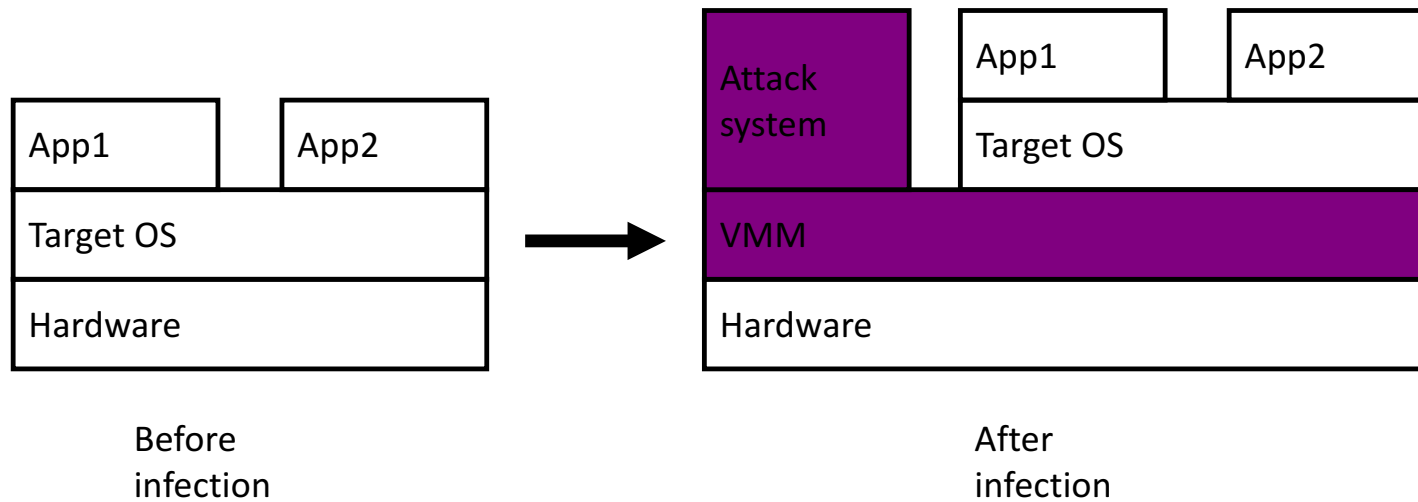
Key: 8x3qrMHjmrH9jfd

Decrypting sector 83234 of 126464 (65%)

Rootkits

- A rootkit modifies the operating system to hide its existence
 - E.g., modifies file system exploration utilities
 - Hard to detect using software that relies on the OS itself
- Operation:
 - Intercept system calls for listing files, processes, etc.
 - Filter out malware's files and processes
 - Example: Magic prefix -- \$sys\$filename
 - Diagram:
 - Applications --> System Call ---> (Rootkit) --> Kernel
 - <--- Results --- If call is from rootkit application (e.g. \$sys\$rootkit.exe), don't filter!

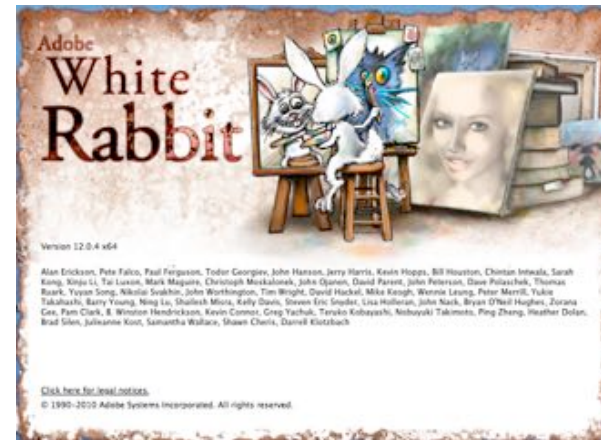
Virtual-machine based rootkits (VMBRs)



Backdoors

- A **backdoor**, which is also sometimes called a **trapdoor**, is a hidden feature or command in a program that allows a user to perform actions he or she would not normally be allowed to do.
- When used in a normal way, this program performs completely as expected and advertised.
- But if the hidden feature is activated, the program does something unexpected, often in violation of security policies, such as performing a privilege escalation.
- Usually enable remote access to the attacker
- Benign example: **Easter Eggs** in DVDs and software

Easter Eggs



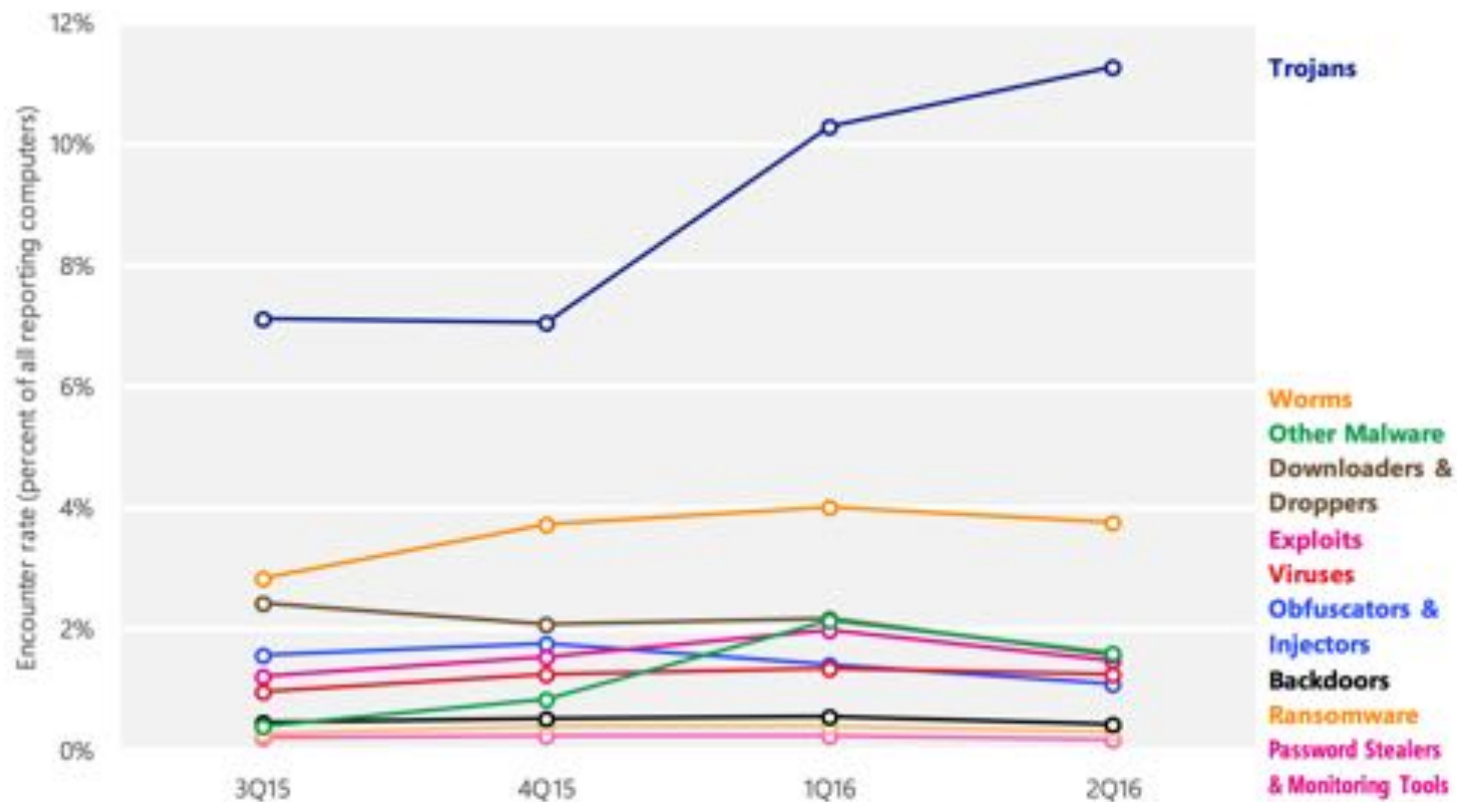
How does malware manage to run?

- Buffer overflow in network-accessible vulnerable service
- Vulnerable client (e.g. browser) connects to remote system that sends over an attack (a driveby)
- Social engineering: trick user into running/installing
- “Autorun” functionality (esp. from plugging in USB device)
- Slipped into a system component (at manufacture; compromise of software provider; substituted via MITM)
- Attacker with local access downloads/runs it directly
 - Might include using a “local root” exploit for privileged access

Insider Attacks

- An **insider attack** is a security breach that is caused or facilitated by someone who is a part of the very organization that controls or builds the asset that should be protected.
- In the case of malware, an insider attack refers to a security hole that is created in a software system by one of its programmers.

Encounter rates for significant malicious software categories, 3Q15–2Q16

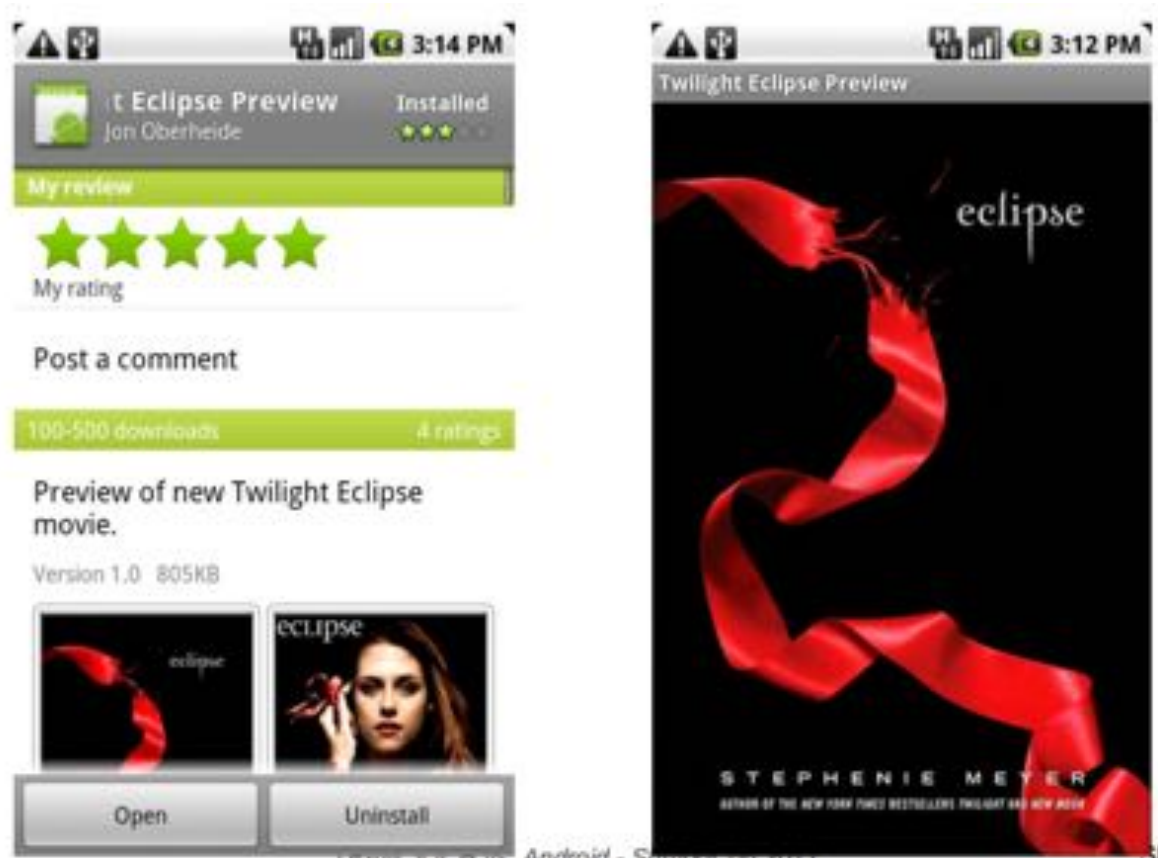


Trojan horse

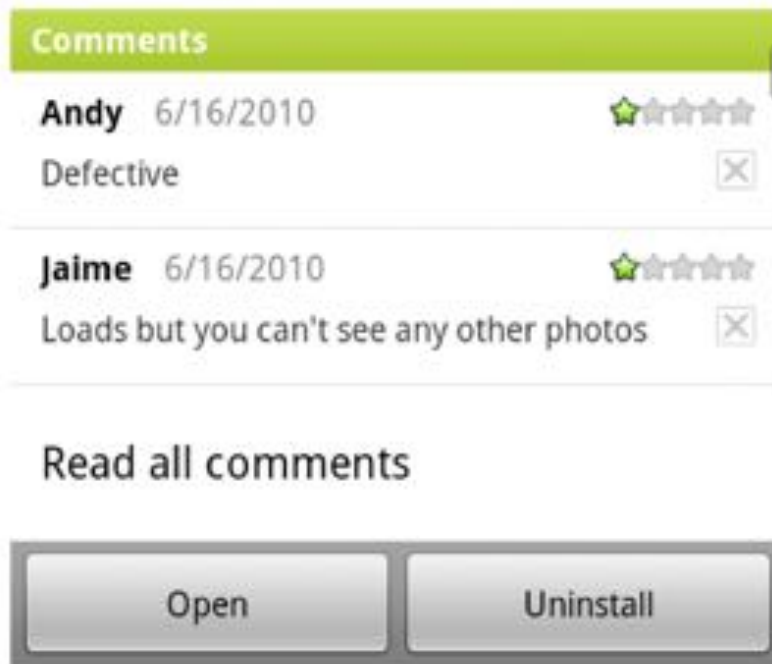
- Software that appears to perform a desirable function but is actually designed to perform undisclosed malicious functions
 - Spyware: installed by legitimate looking programs, then provides remote access to the computer, such as logging keys or sending back documents
 - Adware: shows popup ads
 - Ransomware: encrypts data and requires payment to decrypt



Android Example



Example (cont.)



- Still, 200+ downloads in under 24 hours
- With a legit-looking app/game, you could collect quite an install base for RootStrap

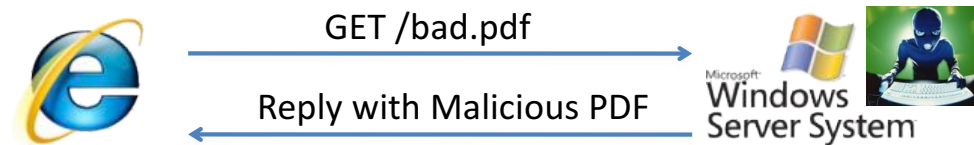


Adware

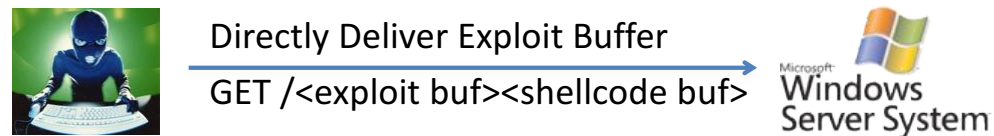


Code Injection Exploits

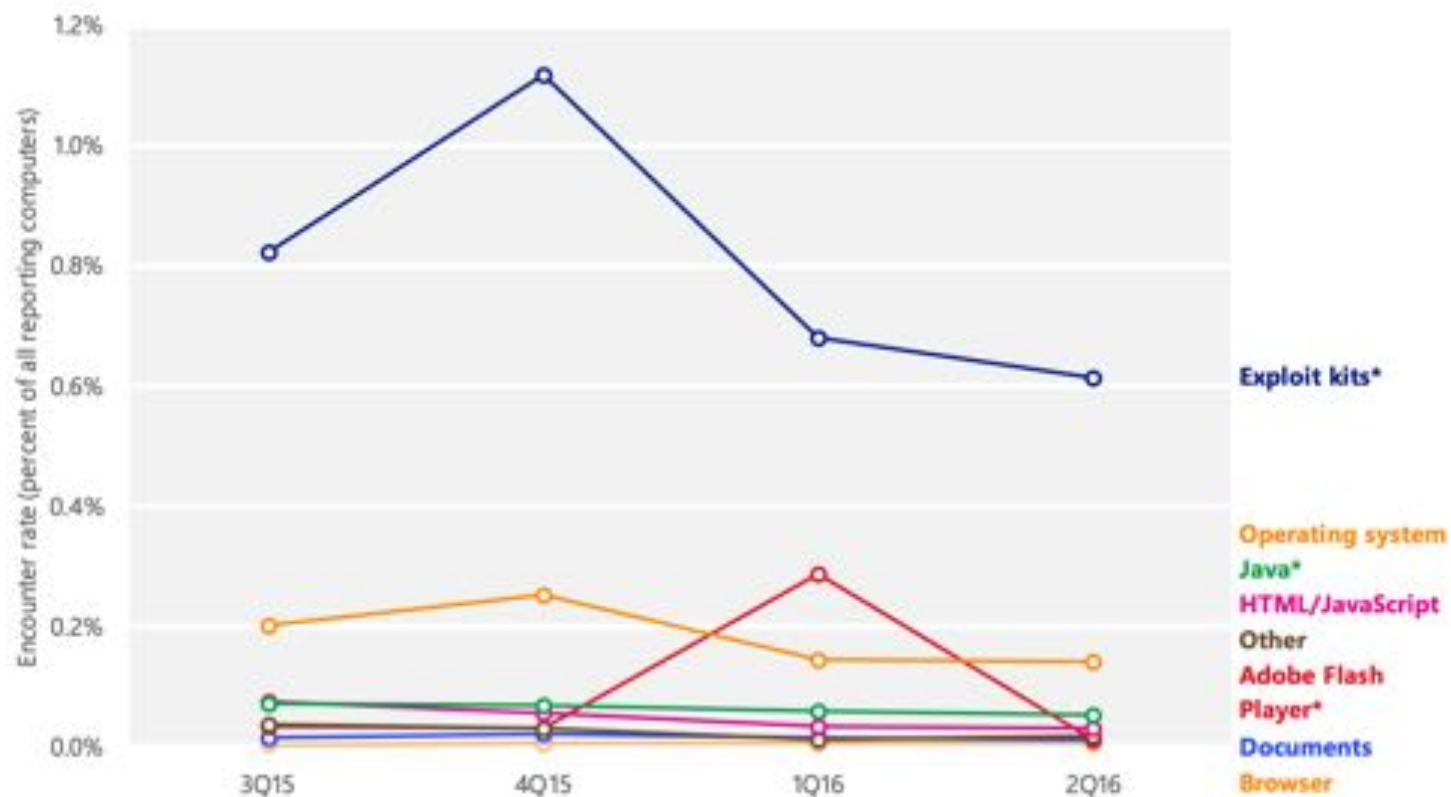
- Client software exploit (e.g. PDF, Flash, MSWord, etc.)



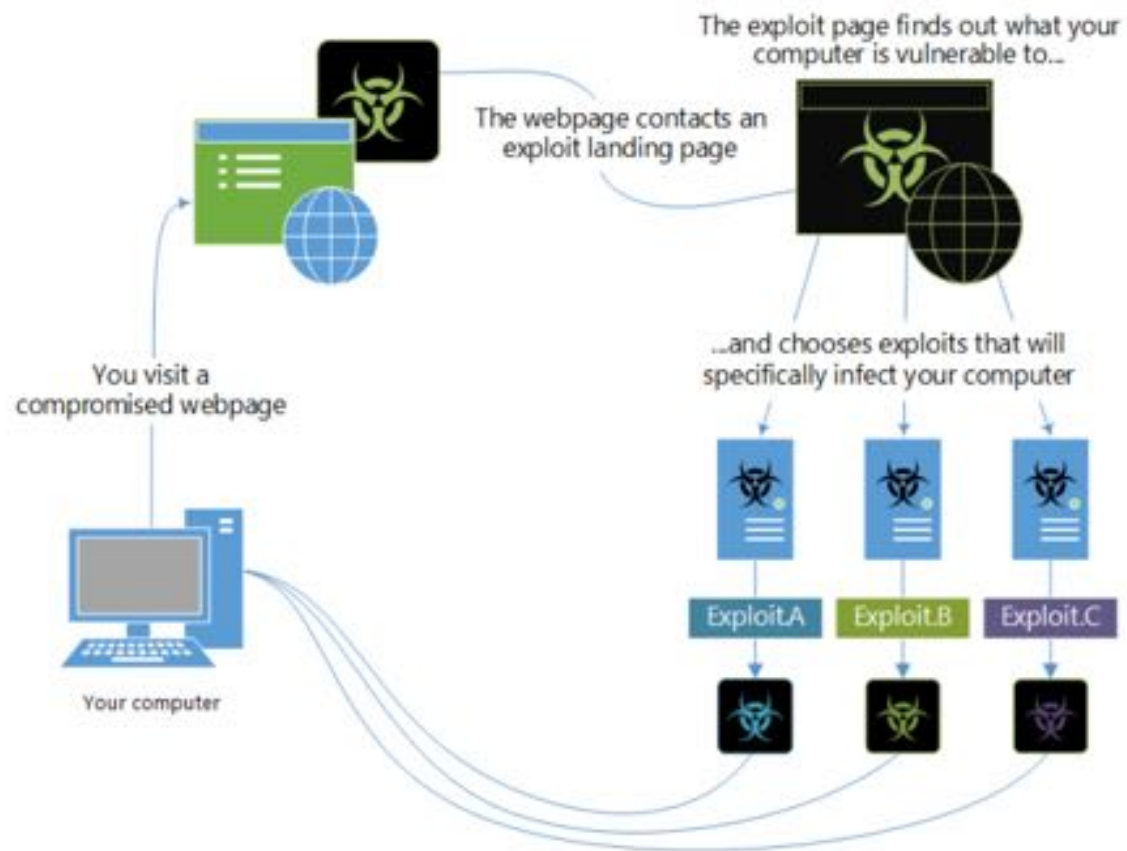
- Network-based exploit (HTTP, File, RPC servers, etc.)



Encounter rates for different types of exploit attempts on the Internet, 3Q15–2Q16



How a typical exploit kit works



Malware That Automatically Propagates

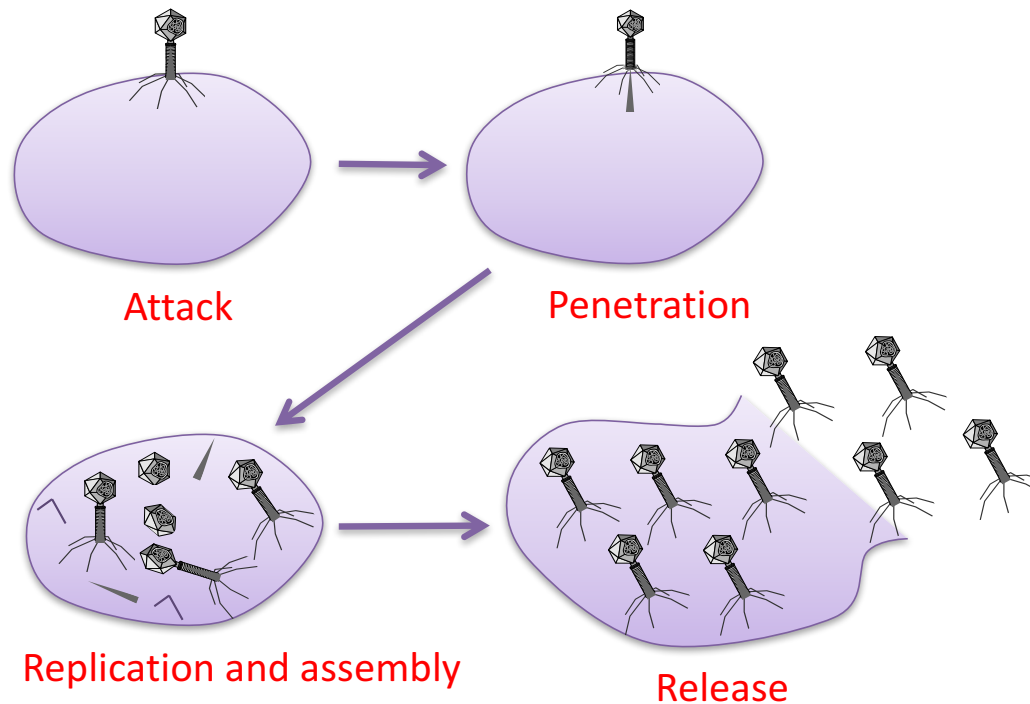
- Virus = code that propagates (**replicates**) across systems by arranging to have itself *eventually executed*, creating additional, new instances of itself
 - Generally infects by altering stored code
 - Typically with the help of a user
- Worm = code that self-propagates/replicates across systems by arranging to have itself *immediately executed*, creating additional, new instances of itself
 - Generally infects by altering running code
 - No user intervention required
- (Note: line between these isn't always so crisp; plus some malware incorporates both styles)

Computer Viruses

- A **computer virus** is computer code that can replicate itself by modifying other files or programs to insert code that is capable of further replication.
- This self-replication property is what distinguishes computer viruses from other kinds of malware, such as logic bombs.
- Another distinguishing property of a virus is that replication requires some type of **user assistance**, such as clicking on an email attachment or sharing a USB drive.

Biological Analogy

- Computer viruses share some properties with Biological viruses



Brain

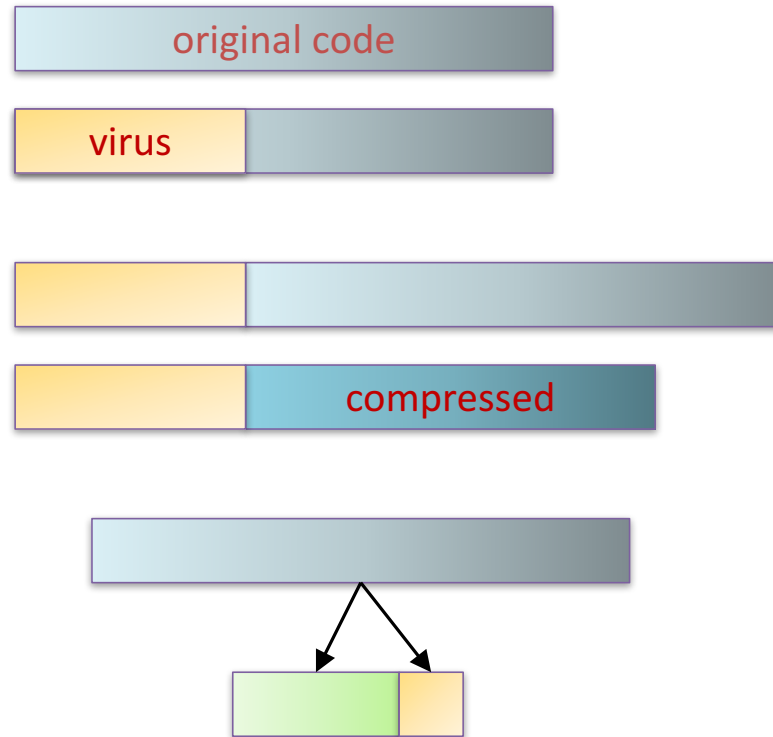
[illegible]

Virus Phases

- **Dormant phase.** During this phase, the virus just exists—the virus is laying low and avoiding detection.
- **Propagation phase.** During this phase, the virus is replicating itself, infecting new files on new systems.
- **Triggering phase.** In this phase, some logical condition causes the virus to move from a dormant or propagation phase to perform its intended action.
- **Action phase.** In this phase, the virus performs the malicious action that it was designed to perform, called its **payload**.
 - This action could include something seemingly innocent, like displaying a silly picture on a computer's screen, or something quite malicious, such as deleting all essential files on the hard drive.

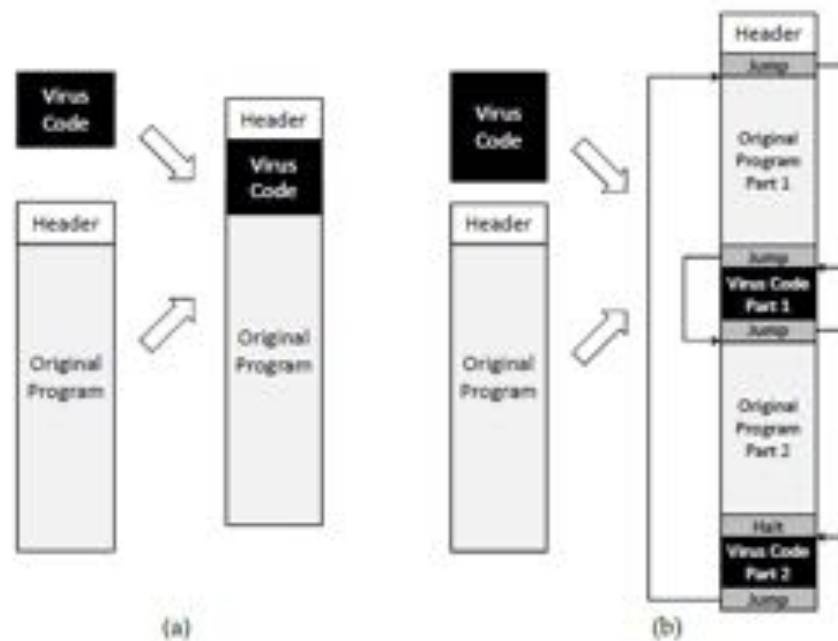
Infection Types

- Overwriting
 - Destroys original code
- Pre-pending
 - Keeps original code, possibly compressed
- Infection of libraries
 - Allows virus to be memory resident
 - E.g., kernel32.dll
- Macro viruses
 - Infects MS Office documents
 - Often installs in main document template
 - LaTeX typesetting system viruses



Degrees of Complication

- Viruses have various degrees of complication in how they can insert themselves in computer code.



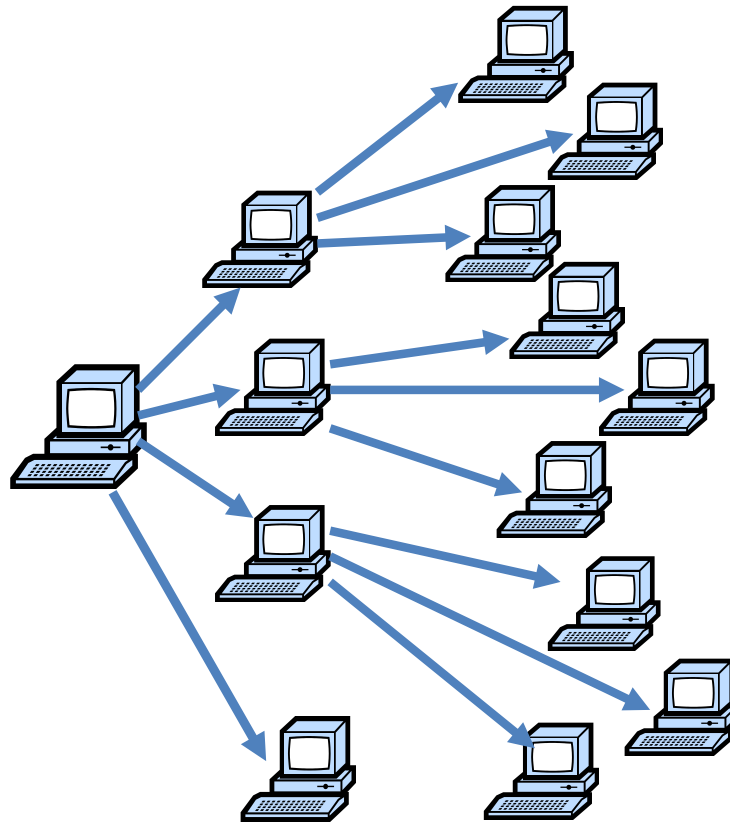
Worm

- Worm = code that self-propagates/replicates across systems by arranging to have itself immediately executed
 - Generally infects machines by altering running code
 - No user intervention required

Rapid Propagation

Worms can potentially spread quickly because they parallelize the process of propagating/replicating.

Same holds for viruses, but they often spread more slowly since they require some sort of user action to trigger each propagation.

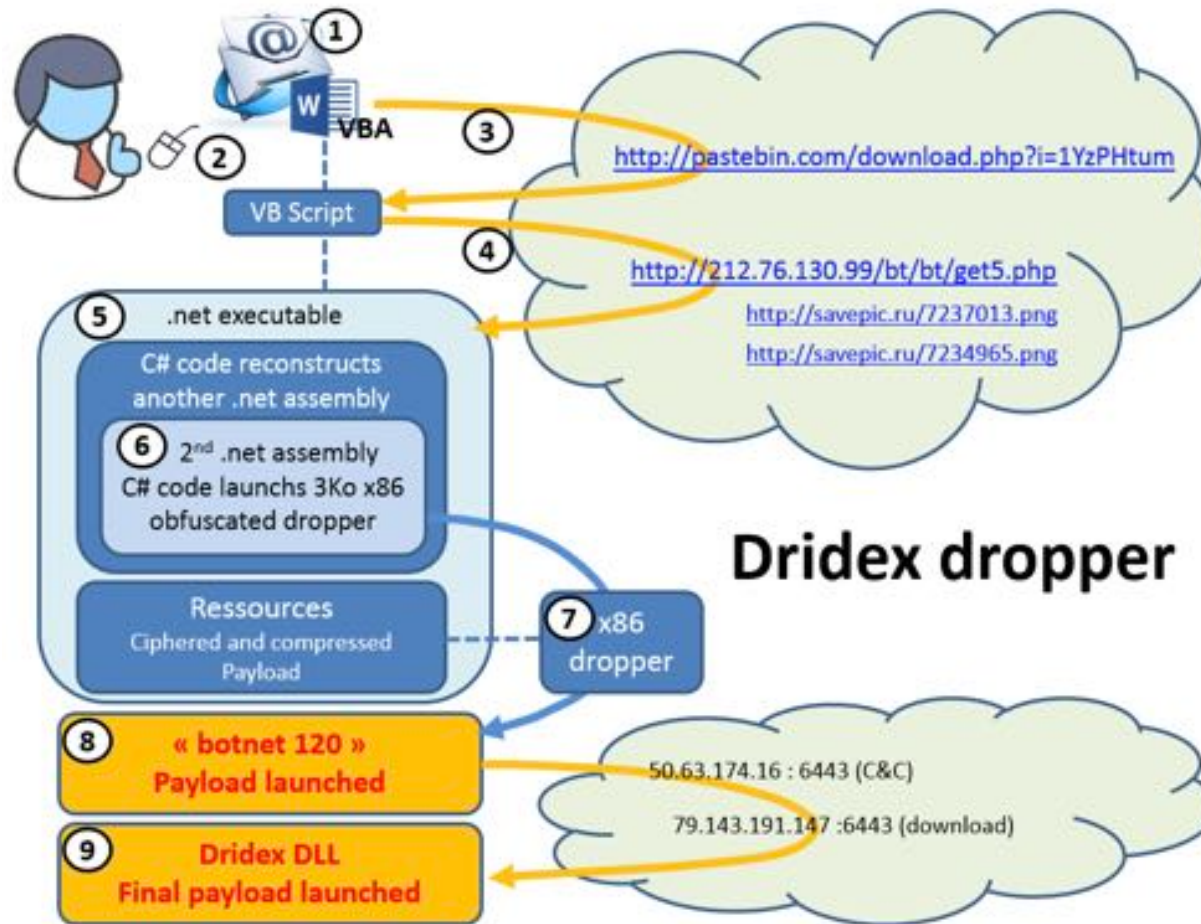


The Arrival of Internet Worms

- Worms date to Nov 2, 1988 - the *Morris Worm*
- **Way** ahead of its time
- Employed a whole suite of tricks to infect systems ...
 - *Multiple* buffer overflows (“gets” function in finger server)
 - Guessable passwords
 - “Debug” configuration option in sendmail that provided shell access
 - Common user accounts across multiple machines
- ... and of tricks to find victims
 - Scan local subnet
 - Machines listed in system’s network config, e.g., /etc/hosts.equiv, /.rhosts
 - Look through user files for mention of remote hosts, e.g., .forward, .rhosts



Droppers



Bridging the how and what of malware: Botnets

- Collection of compromised machines (bots) under (unified) control of an attacker (botmaster)
- Method of compromise decoupled from method of control
 - Launch a worm / virus / drive-by infection / etc.
- Upon infection, new bot “*phones home*” to rendezvous w/ botnet *command-and-control* (**C&C**)
- Lots of ways to architect C&C:
 - Star topology; hierarchical; peer-to-peer
 - Encrypted/stealthy communication
- Botmaster uses C&C to push out commands and updates

Example of C&C Messages

1. Activation (report from bot to botmaster)
2. Email address harvests
3. Spamming instructions
4. Delivery reports
5. DDoS instructions
6. *FastFlux* instructions (rapidly changing DNS)
7. HTTP proxy instructions
8. Sniffed passwords report
9. IFRAME injection/report

From the “Storm”
botnet circa 2008