CS459/698 Privacy, Cryptography, Network and Data Security

Malicious Software

Spring 2025, Monday/Wednesday 2:30pm-3:50pm

Malware?

What is malware?

- Various forms of software written with **malicious intent**
- Common characteristic of all types of malware: needs to be **executed** in order to cause harm
- How might malware get executed?

How might malware get executed?

*…malware needs to be **executed** to cause harm*

- User action
  - Downloading and running malicious software
  - Viewing a web page containing malicious code
  - Opening an executable email attachment
  - Inserting a CD/DVD or USB flash drive
- Exploiting an existing flaw in a system
  - Buffer overflows in network daemons
  - Buffer overflows in email clients or web browsers

Some types of malware

- Viruses
  - Malicious code that adds itself to benign programs/files
  - Code for spreading + code for actual attack
  - Usually activated by users
- Worms
  - Malicious code spreading with no or little user involvement
- Trojans
  - Malicious code hidden in seemingly innocent program that you download
- Botnets
  - Coordinated network of compromised hosts

Viruses

What is a virus?

- A virus is a particular kind of malware that infects other files
  - Traditionally, a virus could infect only executable programs
  - Nowadays, many data document formats can contain executable code
    - Microsoft Word and Excel have macros, PDF files have JavaScript integrations
- Many different types of files can be infected with viruses
- Typically, when the file is executed (or sometimes just opened), the virus activates, and tries to infect other files with copies of itself
- In this way, the virus can spread between files, or between computers

Infection

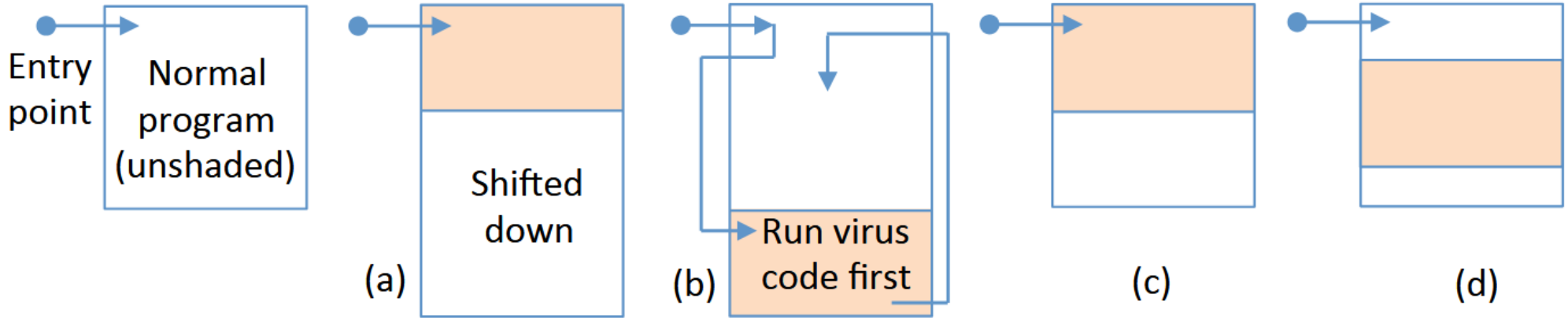
- What does it mean to “infect” a file?
- The virus modifies a (non-malicious) program or document (the host) in such a way that executing or opening it will **transfer control to the virus**
  - The virus can do its “dirty work” and then transfer control back to the host
- For executable programs:
  - Virus will modify other programs and **attach its code to the target**
- For documents with macros:
  - Goal: **add itself as a macro** which automatically runs when file is opened

Strategies for infecting executables

a) Shift and prepend: *original file is executed after virus, file length increased*

b) Append code to end of file: *execution JUMPs to virus code and then back to normal entry point. Evades start of file scans.*

c) Overwrite host file, from top (c) or from interior point (d): *original file destroyed; complicates removal. (d) benefits from avoiding start of file scans*



Computer Security and the Internet: Tools and Jewels from Malware to Bitcoin. Van Oorschot, 2021, p188 (link)

Infection: staying persistent

- In addition to infecting other files, a virus will try to **infect the computer itself**
  - This way, every time the computer is booted, the virus is automatically activated
- Possible examples:
  - Put itself in the boot sector of the hard disk
  - Add itself to the list of programs the OS runs at boot time
  - Infect one or more of the programs the OS runs at boot time
  - It might try many of these strategies
    - But it's still trying to evade detection!

Spreading

- How do viruses spread between computers?
- Usually, when the user sends infected files (hopefully not knowing they're infected!) or compromised website links to his friends
- A virus usually requires some **user action** to spread to another machine
  - If it can spread on its own (via email, for example), it's more likely to be a worm than a virus

Payload

- In addition to trying to spread, what else might a virus try to do?
- Most viruses have some sort of payload (malicious action)
- At some point, the payload of an infected machine will activate, and do something (usually bad)
  - Disable any active virus-scanning software
  - Erase your hard drive, or make your data inaccessible
  - Subtly corrupt some of your spreadsheets
  - Install a keystroke logger to capture your online banking password
  - Start attacking a particular target website

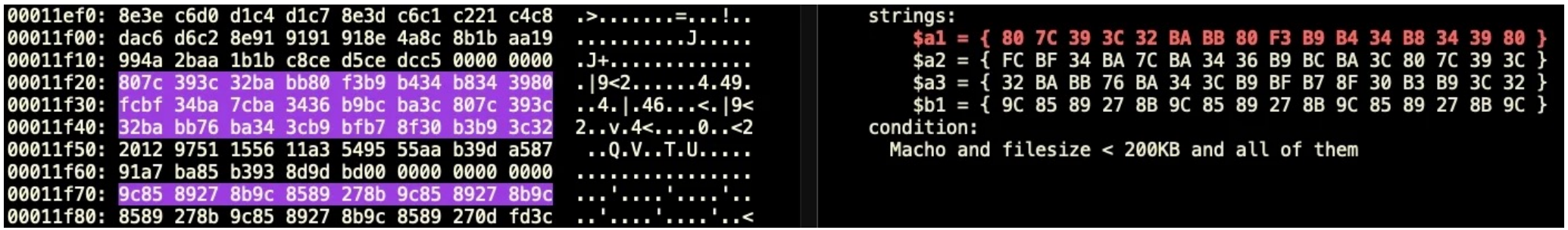
Spotting viruses

When should we look for viruses?

- As files are added to our computer
  - Via portable media
  - Via a network
- From time to time, scan the entire state of the computer
  - To catch anything we might have missed on its way in
  - But of course, any damage the virus might have done may not be reversible
- How do we look for viruses?
  - Signature-based protection and behaviour-based protection

Signature-based protection

- Keep a list of all known viruses
- For each virus in the list, store some characteristic features (the signature)
- Most signature-based systems use features of the **virus code** itself
  - In both the infection code and the payload code
  - Ex) Byte sequences, hashes, code/string matching



Sample of OceanLotus malware and a detection signature for it

Image Source: What is a Malware File Signature?

Polymorphism

- To evade signature-based virus scanners, some viruses are polymorphic
- This means that instead of making perfect copies of itself every time it infects a new file or host, it makes a **modified** copy instead
- Often done by having most of the virus code encrypted
  - The virus starts with a decryption routine which decrypts the rest of the virus, which is then executed
  - When the virus spreads, it encrypts the new copy with a newly chosen random key

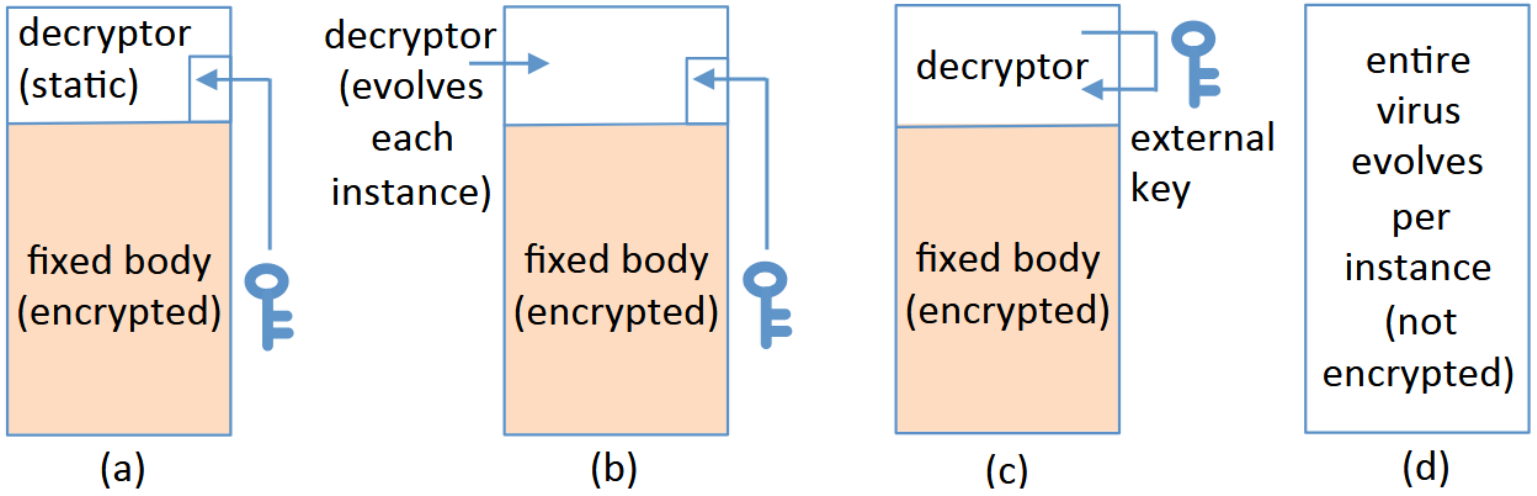
Ways that viruses evade signature detection

a) Virus with encrypted body: *decryptor is static across infections*

b) Mutating encrypted viruses*: changes decryptor and key per infection*

c) Virus with external decryption key: *stores key in another file or machine*

d) Metamorphic virus: *fully rewrites its code with each infection*



Computer Security and the Internet: Tools and Jewels from Malware to Bitcoin. Van Oorschot, 2021, p191 (link)

Downsides of signature-based protection

1. Polymorphic viruses rarely match known signatures
2. You can only scan for viruses that are in the list!
    - But there are brand-new viruses identified **every day**
    - This is another major limitation!

Q: How would you scan for new or polymorphic viruses?

Behaviour-based protection

- Behaviour-based systems look for suspicious patterns of behaviour, rather than for specific code fragments
  - Where on the system it tries to hide itself
  - Unpacking behaviour
  - Accessing specific resources (files, registry keys, API calls)
  - How it propagates from one place to another
- Some systems run suspicious code in a *sandbox* first
  - *Sandbox = secure, isolated environment for executing malware*
- Q: Downsides?

Worms

What is a worm?

- A worm is a self-contained piece of code that can replicate with little or no user involvement
- Worms often use security flaws in widely deployed software as a path to infection
- Typically:
  - A worm exploits a security flaw in some software on your computer, infecting it
  - The worm starts searching for other computers to infect
    - On your local network, or on the Internet generally
  - There may or may not be a payload that activates at a certain time, or by another trigger

The Morris worm, 1988

- The first Internet worm, launched by a graduate student at Cornell
- Once infected, a machine would try to infect others in three ways:
  - Exploit a buffer overflow in the “finger” daemon
  - Use a back door left in the “sendmail” mail daemon
  - Try a “dictionary attack” against local users’ passwords. If successful, log in as them, and spread to other machines they can access without requiring a password
- All three of these attacks were well known!
- First example of buffer overflow exploit in the wild
- Thousands of systems were offline for several days

The Code Red worm, 2001

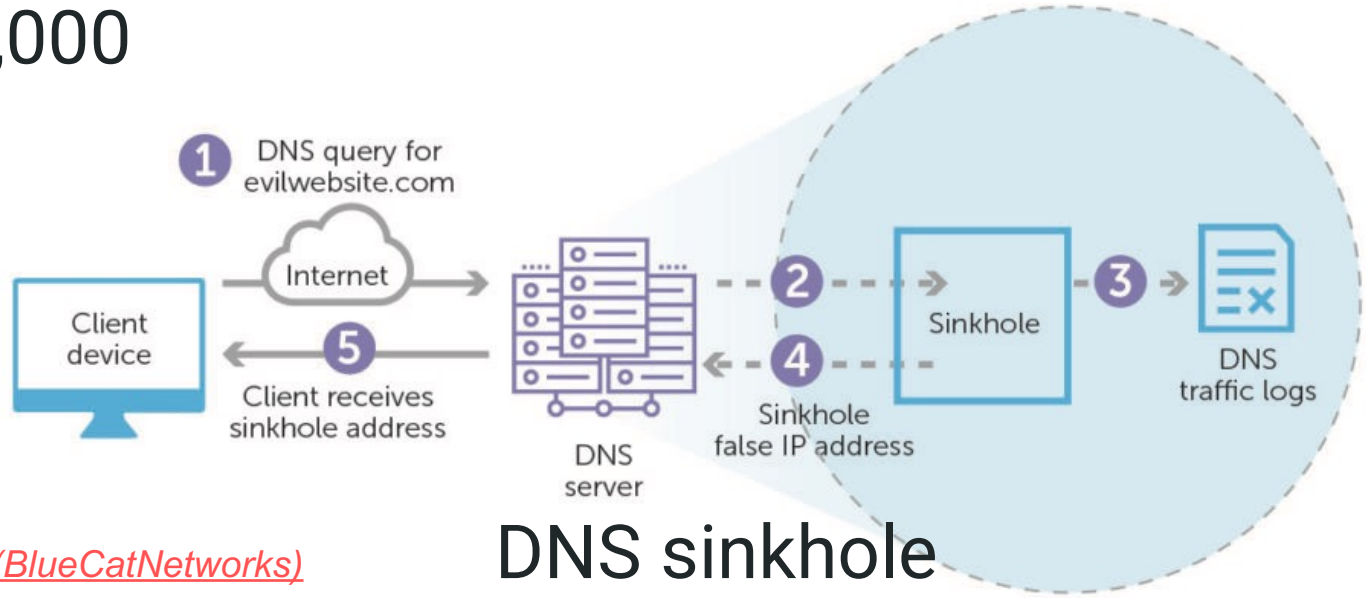
- Exploited a buffer overflow in Microsoft's IIS web server (for which a patch had been available for a month)
- An infected machine would:
  - Deface its home page
  - Launch attacks on other web servers (IIS or not)
  - Launch a denial-of-service attack on a handful of web sites, including www.whitehouse.gov
  - Installed a back door to deter disinfection
- Infected 250,000 systems in nine hours

The Slammer worm, 2003

- Performed **denial-of-service attack**
- First example of a “Warhol worm”
  - A worm which can infect nearly all vulnerable machines in just 15 mins
- Exploited a buffer overflow in Microsoft’s SQL Server (also had a patch available)
- A vulnerable machine could be infected with a **single UDP packet**!
  - This enabled the worm to spread extremely quickly
  - Exponential growth, doubling every **8.5 seconds**
  - 90% of vulnerable hosts infected in 10 minutes

Conficker Worm

- First detected in November 2008
- Propagated a command-and-control style botnet
- Security experts had to generate and sinkhole C&C domains
- Number of infected hosts in 2009: 9–15 million, 2011: 1.7 million, 2015: 400,000



DNS sinkhole

Image Source: DNS Sinkhole (BlueCatNetworks)

Stuxnet

- Discovered in 2010
- Allegedly created by the US and Israeli intelligence agencies
- Allegedly (pretty sure now) targeted Iranian uranium enrichment program
- Targets Siemens SCADA industrial systems installed on Windows
  - One application is the operation of uranium centrifuges
- It tries to be very specific and uses many criteria to select which systems to attack after infection

Stuxnet

- **Very sophisticated**: Used 4(!) different zero-day attacks to spread. Has to be installed manually (USB drive) for air-gapped systems.
- **Very targeted**: Detects if variable-frequency drives are installed, operating between 807–1210 Hz, and then subtly changes the frequencies so that distortion and vibrations occur resulting in broken centrifuges.
- **Very stealthy**: Intercepts commands to SCADA system and hides its presence

Stuxnet

"Iranian President Mahmoud Ahmadinejad observes computer monitors at the Natanz uranium enrichment plant in central Iran, where Stuxnet was believed to have infected PCs and damaged centrifuges." (circa 2008)

https://www.wired.com/2014/11/countdown-to-zero-day-stuxnet/

Worms and the “salami attack”

- A salami attack is made up of smaller, seemingly inconsequential, attacks
- Classic example: send the fractions of cents of round-off error from many accounts to a single account owned by the attacker
- More commonly:
  - Credit card thieves make very small charges to very many cards
  - Clerks slightly overcharge customers for merchandise
  - Gas pumps misreport the amount of gas dispensed

Trojans

Have you ever seen this?

http://www.sampsonuk.net/B3TA/TrojanHorse.jpg

What are Trojan Horses?

- Trojan horses are programs which claim to do something innocuous (and usually do), but which also hide malicious behaviour

"You're surfing the Web and you see a button on the Web site saying, "Click here to see the dancing pigs." And you click on the Web site and then this window comes up saying, "Warning: this is an untrusted Java applet. It might damage your system. Do you want to continue? Yes/No."
Well, the ***average computer user is going to pick dancing pigs over security any day****. And we can't expect them not to."* — Bruce Schneier

Dancing Pig

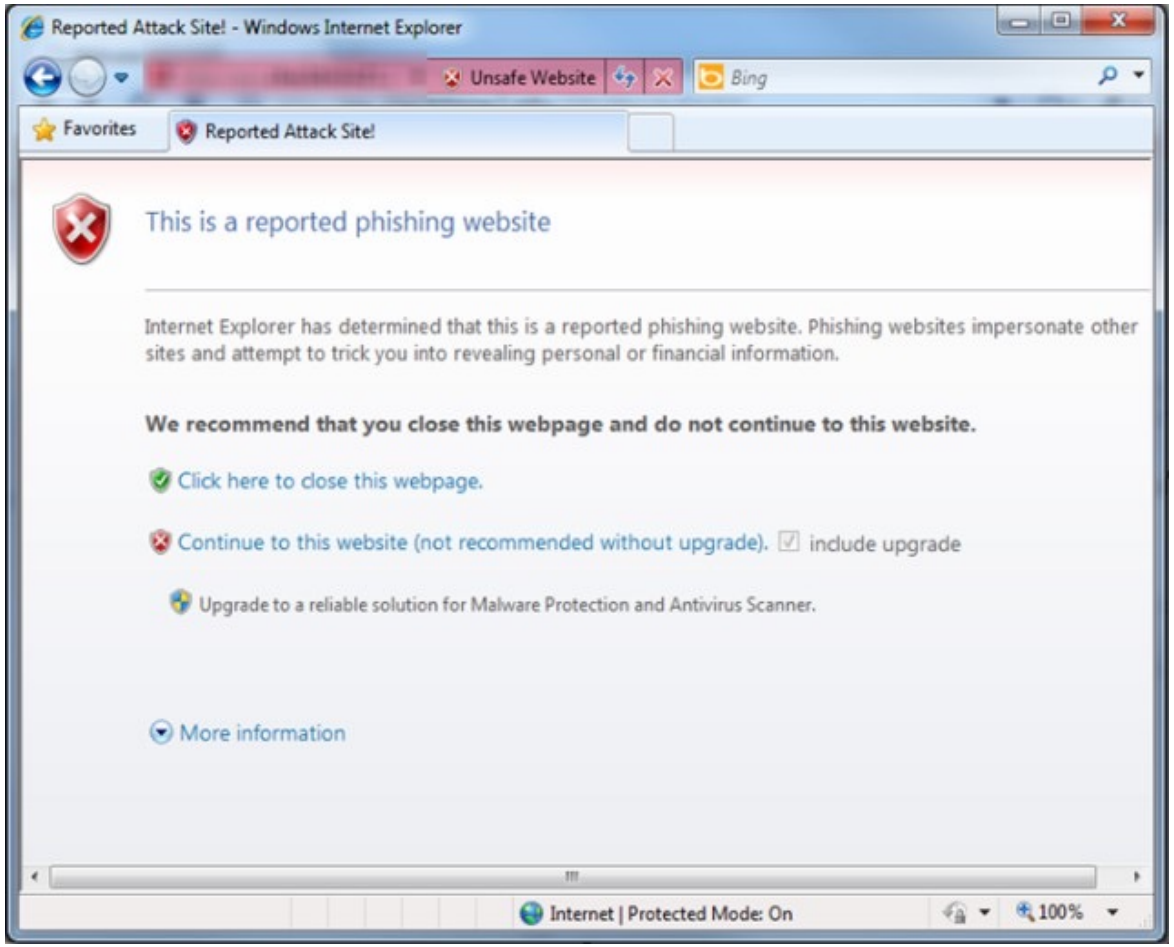
“You’re surfing the Web and you see a button on the Web site saying, “Click here to see the dancing pigs.” And you click on the Web site and then this window comes up saying, “Warning: this is an untrusted Java applet. It might damage your system. Do you want to continue? Yes/No.”
Well, the ***average computer user is going to pick dancing pigs over security any day****. And we can’t expect them not to.”* — Bruce Schneier

Image Source: Dancing Happy Pig

How do Trojan Horses work?

- Gain control by getting the user to run code of the attacker's choice, usually by also providing some code the user **wants** to run
  - "PUP" (potentially unwanted programs) are an example
  - For scareware, the user might even pay the attacker to run the code
- The payload can be anything; sometimes the payload of a Trojan horse is itself a virus, for example
- Trojan horses usually do not themselves spread between computers; they rely on multiple users executing the "trojaned" software

Scareware



http://static.arstechnica.com/malware_warning_2010.png

Ransomware

https://en.wikipedia.org/wiki/WannaCry_ransomware_attack#/media/File:Wana_Decrypt0r_screenshot.png

How does ransomware work?

- Demands ransom to return some hostage resource to the victim

- CryptoLocker in 2013:
  - Spread with spoofed e-mail attachments from a botnet
  - **Encrypted** victim's hard drive
  - Demanded ransom for **private key**
  - Botnet taken down in 2014; estimated ransom collected between $3 million to $30 million

- Could also be scareware

WannaCry

- Ransomware launched in May 2017
- Infected ~230,000 computers
- Exploits a Windows SMB vulnerability originally discovered by the NSA
- NSA kept it secret (and exploited it)
- The “Shadow Brokers” leaked it (and others) in April 2017
- Microsoft had released a patch after being alerted by NSA but many systems remained unpatched
- Emergency patch for Windows XP and 8 in May 2017

See extra slide for more information on WannaCry’s Cryptographic Mechanism

WannaCry


NBC NEWS
POLITICS
U.S. NEWS
WORLD
BUSINESS
SHOPPING
SPORTS
HEALTH
NBC NEWS TIPLINE
WATCH LIVE
WORLD
Why 'WannaCry' Malware Caused Chaos for National Health Service in U.K.
The cyberattack has quickly become another political football in the years-long battle over the funding, remit, and existential future of the NHS.

Botnets

Motivations for building botnets

- Earlier worms (Nimda, Slammer) were written by hackers for fame with the goal to spread worm as fast as possible
  - Caused disruption and helped detection
- Today's botnets are controlled by crackers looking **for profit**, which rent them
  - Criminal organizations
- Can spread more slowly and in targeted ways
  - **Intelligence** and **espionage**?

New Generation Botnets

- Today's botnets are very sophisticated
- Virus/worm/trojan for propagation, exploit multiple vulnerabilities
- Stealthiness to hide from owner of computer
- Code morphing to make detection difficult
- Bot usable for different attacks (spam, DDoS, ...)

Botnet's infrastructure

- Distributed, dynamic & redundant control infrastructure
- "Fast Flux"
  - A single host name maps to hundreds of addresses
  - Machines are constantly swapped in/out of DNS to make tracking difficult
- "Domain Flux" (w/ Domain Generation Algorithm)
  - Infected machine generates a large set (50,000 in the case of Conficker) of domain names that changes every day, contacts a random subset of these names for updates
  - To control the botnet, authorities would have to take control of 50,000 different domain names each day – see optional reading!

Sample botnet: Storm

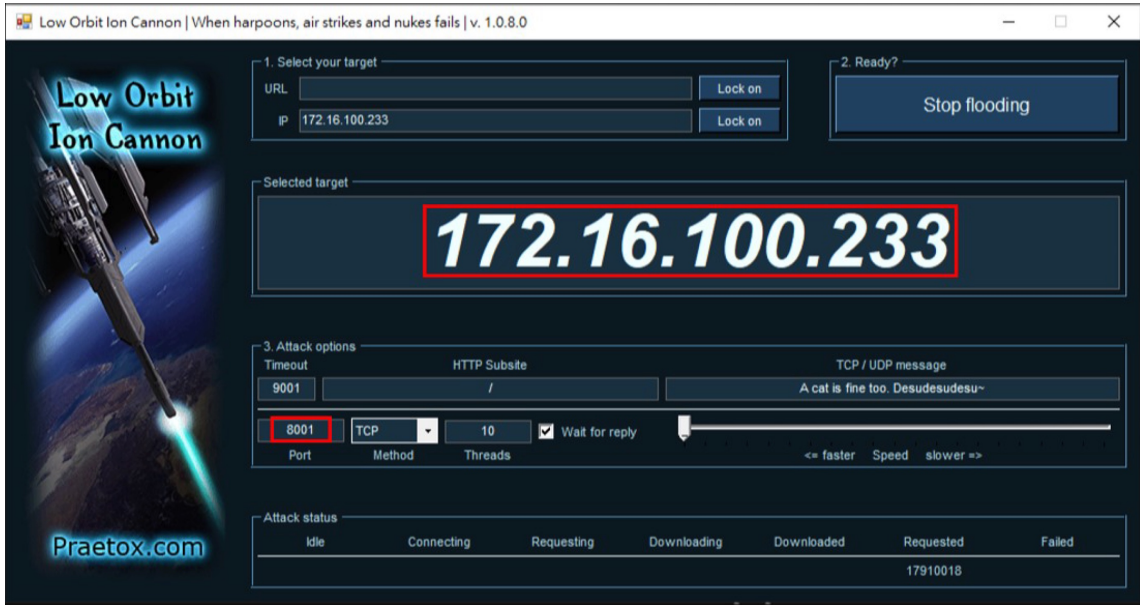
- In September 2007, Storm Worm botnet included hundreds of thousands or even millions of machines
- Bots were used to send out junk emails advertising web links that when clicked attempted to download and install worm, or to host these websites
- Botnet was also rented out for pharmacy and investment spam
- As a self-defence mechanism, it ran DDoS attacks against Internet addresses that scanned for it
- Problem: its P2P protocol created >10 times usual P2P traffic (=> detectable)

Sample botnet: Mirai

- In Fall 2016, the Mirai botnet attacked several high-profile targets, including a popular security blog, a large DNS provider, and Liberia
- Attack traffic of so far unseen **1 Tbps** or more
  - (previous DDoS attacks were 50 Gbps)
- Botnet consisted of **600,000 IoT devices** (routers, cameras) infected due to **unchanged default passwords**
- Distribution based on self-propagating worm
- Each bot flooded targets with UDP, TCP, and HTTP traffic
- Botnet developed by a Minecraft server DDoS protection company

The new script kiddie on the block

- All the discussed attacks, exploit code and complete attack scripts are available on the Internet
- Script kiddies can download scripts and raise an attack with minimum effort
- There are even tools that allow easy building of individual attacks:
  - E.g., Metasploit Framework, based on existing exploits
  - E.g., LOIC, stress testing and denial-of-service



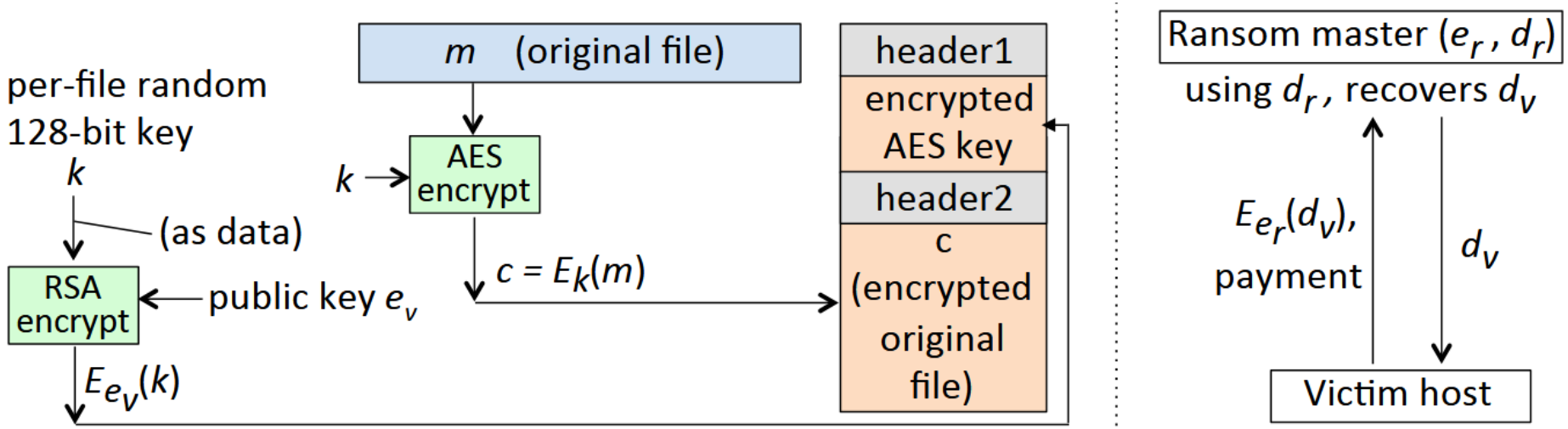
Extra slides (interesting info)

Cryptographic mechanism for WannaCry

- Asymmetric cryptography is slow to encrypt … might take too long
  - Will also create a file that's larger than original (ciphertext much larger than plaintext)
- Usually, for encrypting files they used block ciphers like AES
  - They can also encrypt the first few bytes for the file
  - Usually, the rest of the file would be unreadable
  - Efficient!
- WannaCry uses per-file random key
  - Keys are encrypted using public key of the attacker
  - Put encrypted key in the file itself

Cryptographic mechanism for WannaCry

1) Ransomware master possesses public/private key pair (e_r, d_r)
2) Generates per-victim key pair (e_v, d_v), but encrypts victim private key using master public key $E_{e_r}(d_v)$
3) Generates random AES key k for every file, and encrypts file
4) Every k gets encrypted $E_{e_v}(k)$ and placed at the start of each file



Computer Security and the Internet: Tools and Jewels from Malware to Bitcoin. Van Oorschot, 2021, p203 (link)