Reverse Engineering: Towards Malware Analysis

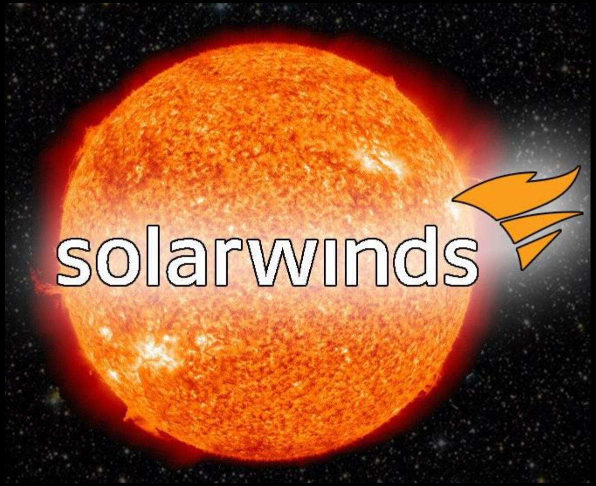
Lecture - Introduction

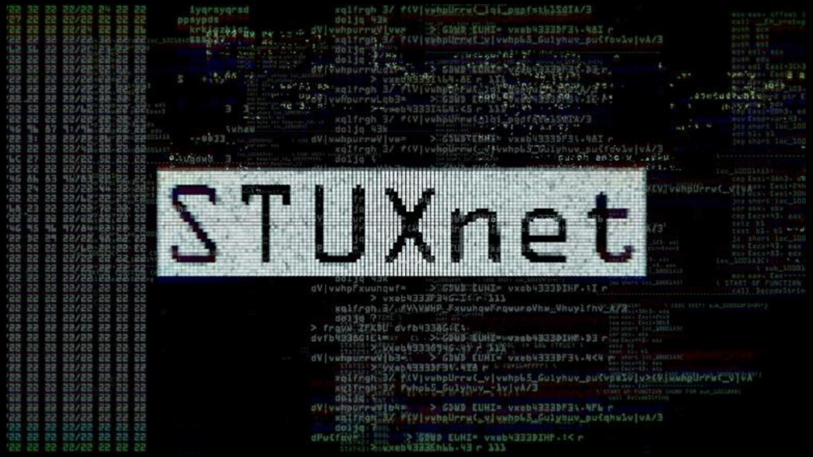
Computer Security Practice

NotPetya
Ransomware

Wanna Cry

Conficker


solarwinds


STUXnet

Grading

Two components:

- Homework (2 assignments)
- Project

Acknowledgements

Shout out and special thanks to:

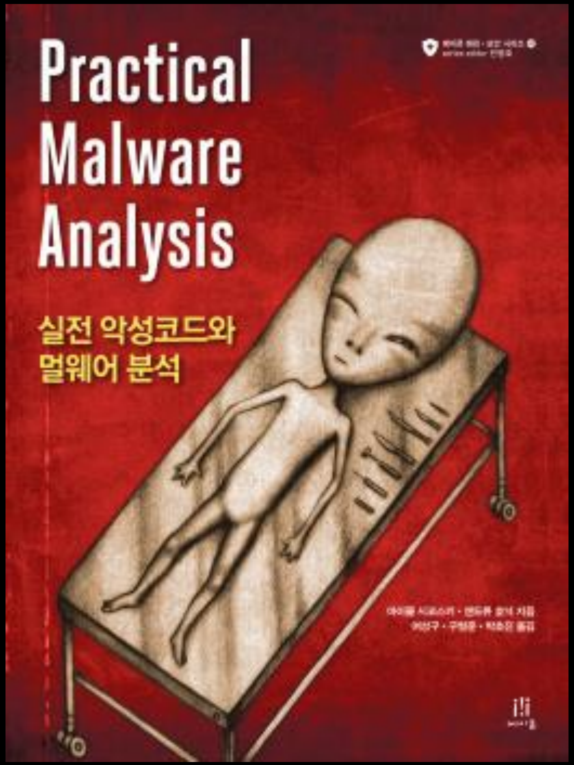
Michael Sikorski, Vice President of FLARE Team @ FireEye

With his permission, this lecture segment was adapted from:

E6998 - Malware Analysis and Reverse Engineering
Columbia University – Computer Science

Textbook: No Starch Press

no starch
press



Overall Goals

- Learn about reverse engineering by analyzing malware
- Build skills so you can have success on your own
  - Hands-on learning
- Practical!
- Having fun with puzzles

Reverse Engineering

Software

- Cracking
- Pirating
- Vulnerability Research
- Malware Analysis

Hardware

- Legacy
- Stealing IP

Malware

- Any code that performs evil (“malicious software”)
- Unknown code on machine of interest
- Types
- Rootkit, Backdoor, Botnet, Scareware, Worm, Virus
- Lack of understanding = Lack of capability

What is Malware Analysis?

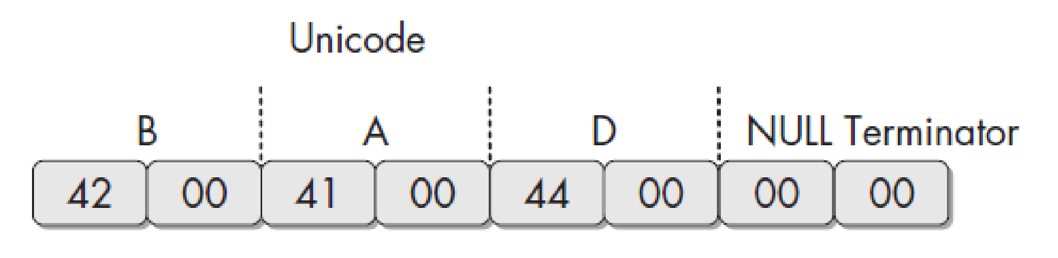
- Dissecting malicious software
- Cat and mouse game
- It is NOT:
  - Finding malware
    - Forensics
    - IR
- Drives Incident Response (if done right)
  - Security is better with malware analysis
    - Host-Based Indicators
    - Network Signatures

Analysis Techniques

- Basic Static – Autopsy
    - Checksums
    - Strings
    - PE File Format
- Basic Dynamic – Running
    - Monitoring tools
    - Faking the network
- Advanced – Reverse Engineering
    - Disassembly – looking at the code before it’s run
    - Debugging – running the code and observing it’s internal state

Tip toe in!

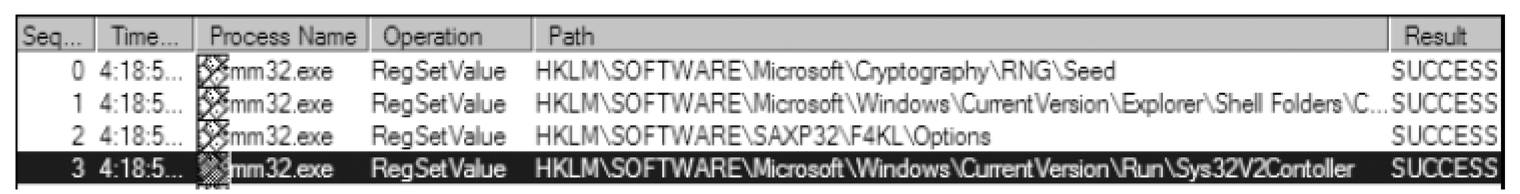
- Chapters 0-3
  - Focus of most introduction classes
  - Basic Static



Unicode

B		A		D		NULL Terminator	
42	00	41	00	44	00	00	00

  - Dynamic Tools



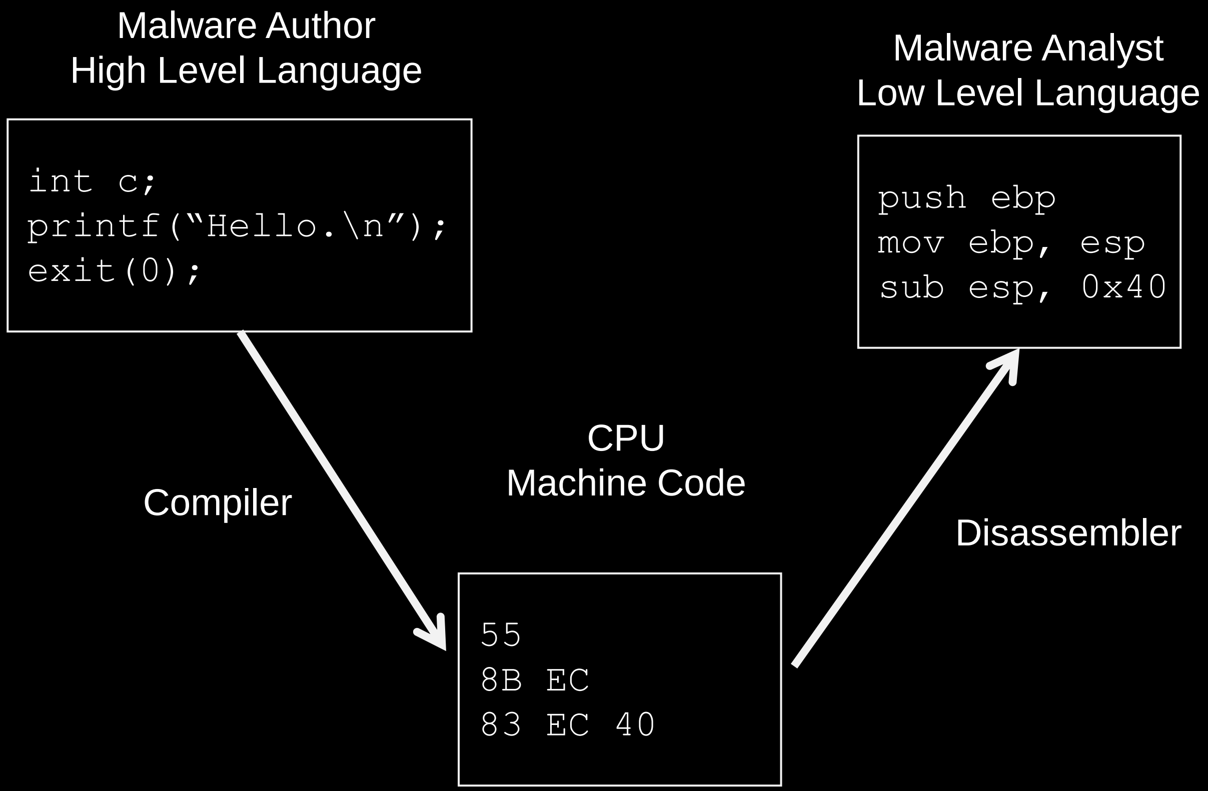
Seq...	Time...	Process Name	Operation	Path	Result
0	4:18:5...	mm32.exe	RegSetValue	HKLM\SOFTWARE\Microsoft\Cryptography\RNG\Seed	SUCCESS
1	4:18:5...	mm32.exe	RegSetValue	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders\C...	SUCCESS
2	4:18:5...	mm32.exe	RegSetValue	HKLM\SOFTWARE\SAXP32\F4KL\Options	SUCCESS
3	4:18:5...	mm32.exe	RegSetValue	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\Sys32V2Contoller	SUCCESS

  - Creating a safe environment
  - Gentle introduction for all!
  - Should be baseline for incident responders

Cliff Dive!

- Cliff dive with Cliff notes parachute
  - Build the foundation

- Chapter 4-9
  - Assembly
  - Windows
  - Debuggers
  - Disassemblers
  - Kernel space



Malware Author
High Level Language

```
int c;
printf(“Hello.\n”);
exit(0);
```

Compiler

CPU
Machine Code

```
55
8B EC
83 EC 40
```

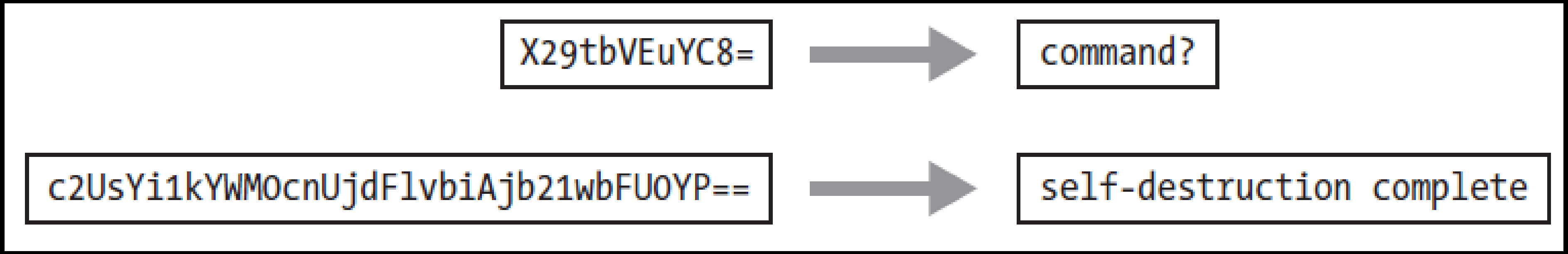
Disassembler

Malware Analyst
Low Level Language

```
push ebp
mov ebp, esp
sub esp, 0x40
```

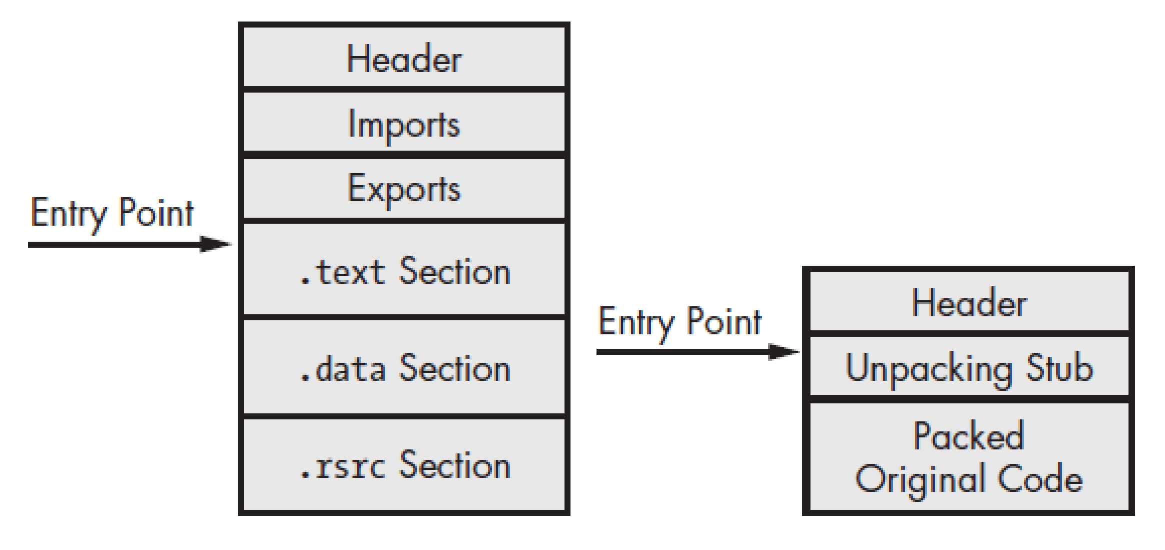
Malware Functionality

- Chapters 11-14
  - Behavior
  - Covert launching
  - Network Signatures *<not covered in this course>*
  - Encoding and Encryption *<not covered in this course>*

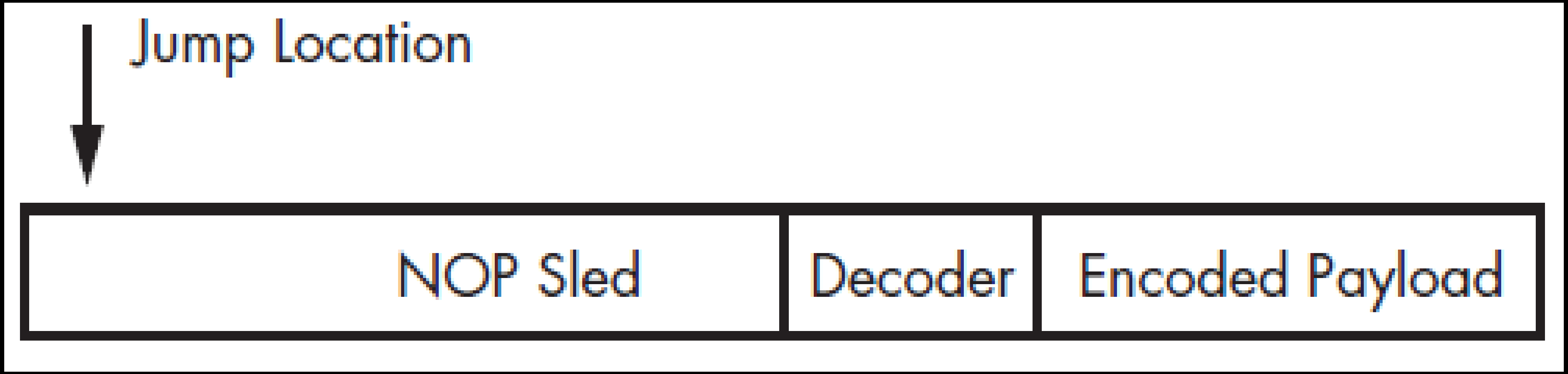

X29tbVEuYC8=
command?
c2UsYi1kYWMOcnUjdFlvbiAjb21wbFUOYP==
self-destruction complete

Anti-Reversing

- Chapter 15-18 *<not covered in this course>*
  - Malware authors thwart your analysis
    - Anti-debugging
    - Anti-disassembly
    - Anti-VM
    - Packing
  - Learn to fight back



Shellcode



Malware Analysis Assist

- Machine Learning
  - Clustering
    - Big variant problem
    - Clients with a lot of binaries
  - Classifying
    - Good or bad
    - Seen before

Book Appendix Resources

- Tools review
  - No extensive list of tools exists
    - Many do the same thing
    - Let you decide
- Windows API cheat sheet
  - Meant for the beginner
  - Focus on common and non-obvious functions

Goals of Malware Analysis

- Host-Based Indicators
    - What malware does to a system
    - Better than traditional signatures
        - Lasts longer
        - Variants
- Network-Based Signatures
    - Finding malware on a network
    - Better with reversing engineering than just traffic analysis alone
- Full Report
    - Exactly what the malware does

Host-Based Signatures

Finding malware on the host

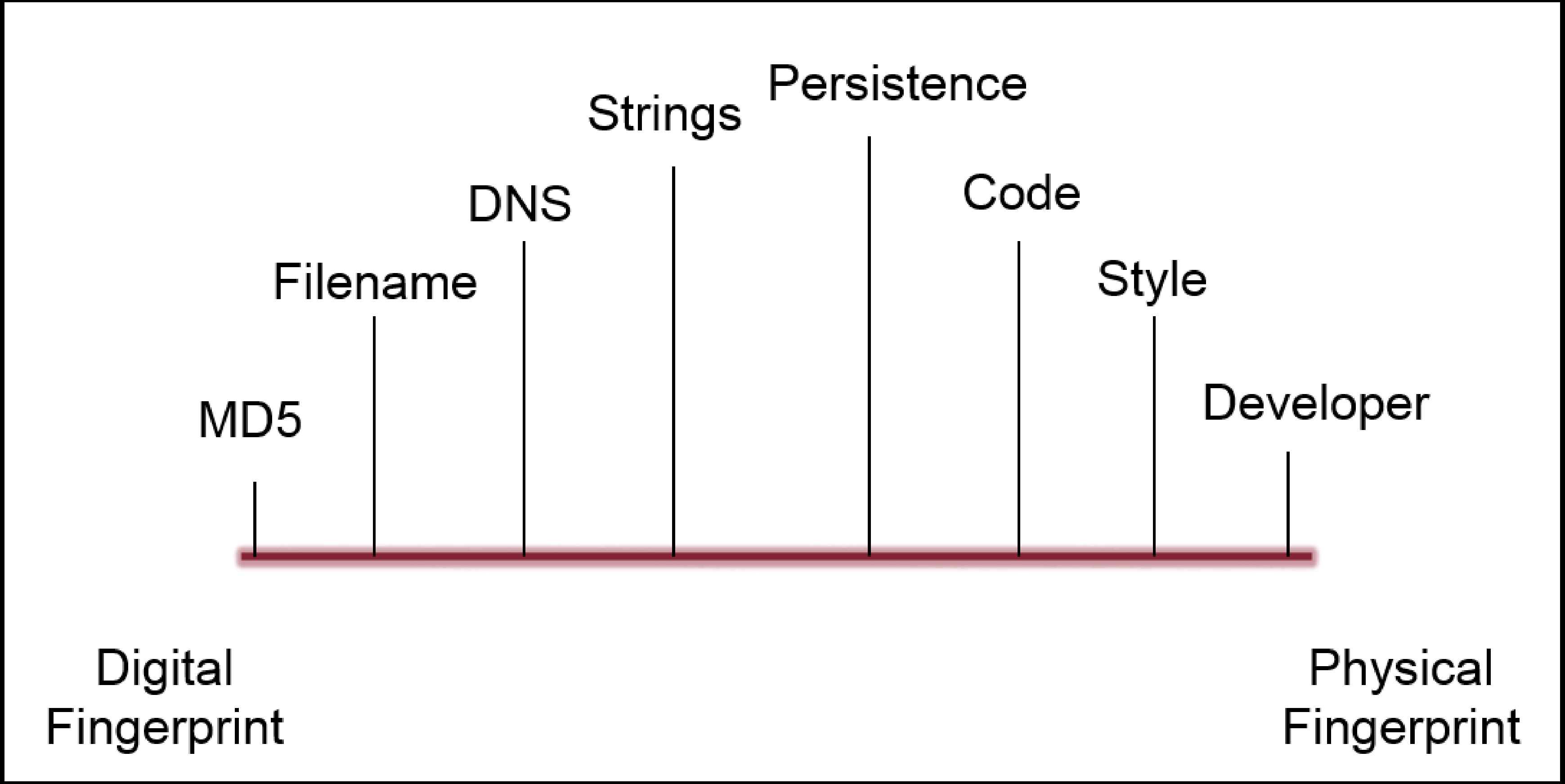
- Files system changes – file creation, modification, etc
- Registry changes
- Persistence mechanisms
- Volatile Evidence

Network-Based Signatures

Finding malware on the network

- IP address
- Domain Name
- Snort Signature

Signatures as Fingerprints



General “Golden” Rules

- Don’t get caught in the details
- Focus on key features
- Get general overview
- Different Tools and Approaches
  - Step back try different angle
- Cat-and-mouse game
- There is no cheating in malware analysis
- Don’t get caught in the details
- I will repeat this again and again

Other Admin

- Schedule
- Homework
- Grading

Schedule

Week No	Lecture	Topic Covered	Reading	Homework	Project / Lab
Week 8	A	Intro, Basic Static Analysis, PE File Format Explained	Chap 0, 1	HW1 (due Week 10)	
Week 9	B	Safe Environment, Basic Dynamic Analysis	Chap 2, 3		
Week 10	C	x86 Disassembly, IDA Pro, C Constructs	Chap 4, 5, 6	HW2 (due Week 12)	
Week 11	D	Windows Internals and Malware	Chap 7		Lab 1 (due Week 14)
Week 12	E	Debugging, OllyDbg	Chap 8, 9	HW3 (due Week 14)	
Week 13	F	Covert Malware Launching, Malware Behaviour	Chap 11, 12		
-					

WARNING!!!

Be careful

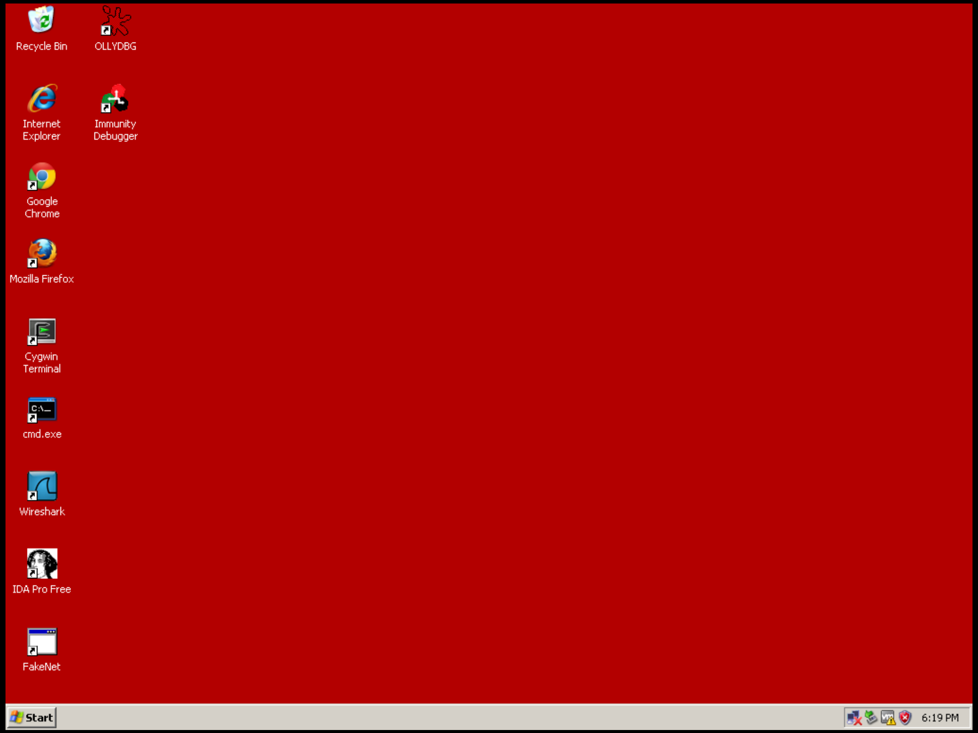
- Always store and transport suspected malware in a password protected and encrypted archive, such as ZIP or RAR. Use a common password such as “infected”; not to protect the contents, but to prevent accidental examination or execution. This prevents antivirus and other security products from removing or quarantining the malware.

- Homework password – “infectednus”

- Use a modern version of your primary operating system and ensure that system is fully patched and updated.
- Ensure that all third party software is fully patched and updated.
- Disable preview views.
- Disable autorun or automount features.
- Handle suspected malware while logged in as a non-privileged user.
- Add an underscore to the end of suspected malware file extensions or renamed the extension to help prevent accidental execution or opening. For example change the extension “.exe” into “.exe_”.
- Store all suspected malware and malware archives in a directory that denies execution, and only allows access to a non-privileged user.
- Do not access or launch suspected malware unless you are operating in a virtual machine or other isolated analysis environment. The exceptions are to initially save the file to your hard drive, create the storage archive, or transfer the archive to your analysis environment. We even recommend avoiding the computation of a checksum or viewing strings unless you are in a virtual environment. You should develop the natural habit of only working with malware in the virtual environment.

Virtual Machine

- What is a VM?
- Windows XP / 10
  - Your Safe Environment
- Take frequent snapshots
- We will use Flare-VM
- Delete after class
  - *Use for education only*



?