Reverse Engineering

Malware and Adversarial Modeling

Spring 24
Joshua Reynolds
NMSU

Reverse Engineering

Taking something apart to figure out how it works or is made

In software this can be used to:

1. Steal technology/IP (e.g. patent infringement)
2. Avoid DRM (think ‘cracked’, pirated software or VLC)
3. Make compatible products (like how LibreOffice can open .docx files)
4. Figure out what malicious programs do, and how to stop them
5. Develop software exploits

Reverse Engineering

Taking something apart to figure out how it works or is made

In software this can be used to:

1. Steal technology/IP (e.g. patent infringement)
2. Avoid DRM (think ‘cracked’, pirated software or VLC)
3. Make compatible products (like how LibreOffice can open .docx files)
4. **Figure out what malicious programs do, and how to stop them**
5. Develop software exploits

We will study #4 in this course.

Definitions

Malware

Any malicious program

Virus

A malicious program whose spread requires human help (running a program, opening a PDF, plugging a USB, etc.)

Worm

A malicious program whose spread is automatic (like Mirai)

Who writes malware?

Individuals, groups, countries, (AI soon?)

Who writes malware?

We can classify malware by who has written it.

It will have different goals based on why it was created.

Who writes malware?

We can classify malware by who has written it.

It will have different goals based on why it was created.

Nation States

- Cozy Bear - Russia
- Lazarus - N. Korea
- NSA, CIA - USA
- Mossad - Israel
- ICA - Iran
- Double Dragon - China

Who writes malware?

We can classify malware by who has written it.

It will have different goals based on why it was created.

Cyber Criminals

- Access as a service (AAAS)
- Ransomware, and Ransomware as a service (RAAS)
- Exploit development & sale
- Bulletproof hosting
- Phishing as a Service (PAAS)
- Booter/Stresser, DDoS as a Service (DAAS)

Who writes malware?

We can classify malware by who has written it.

It will have different goals based on why it was created.

Hacktivists

- Anonymous
- LulzSec
- Chaos Computer Club
- Lizard Squad
- Legion of Doom

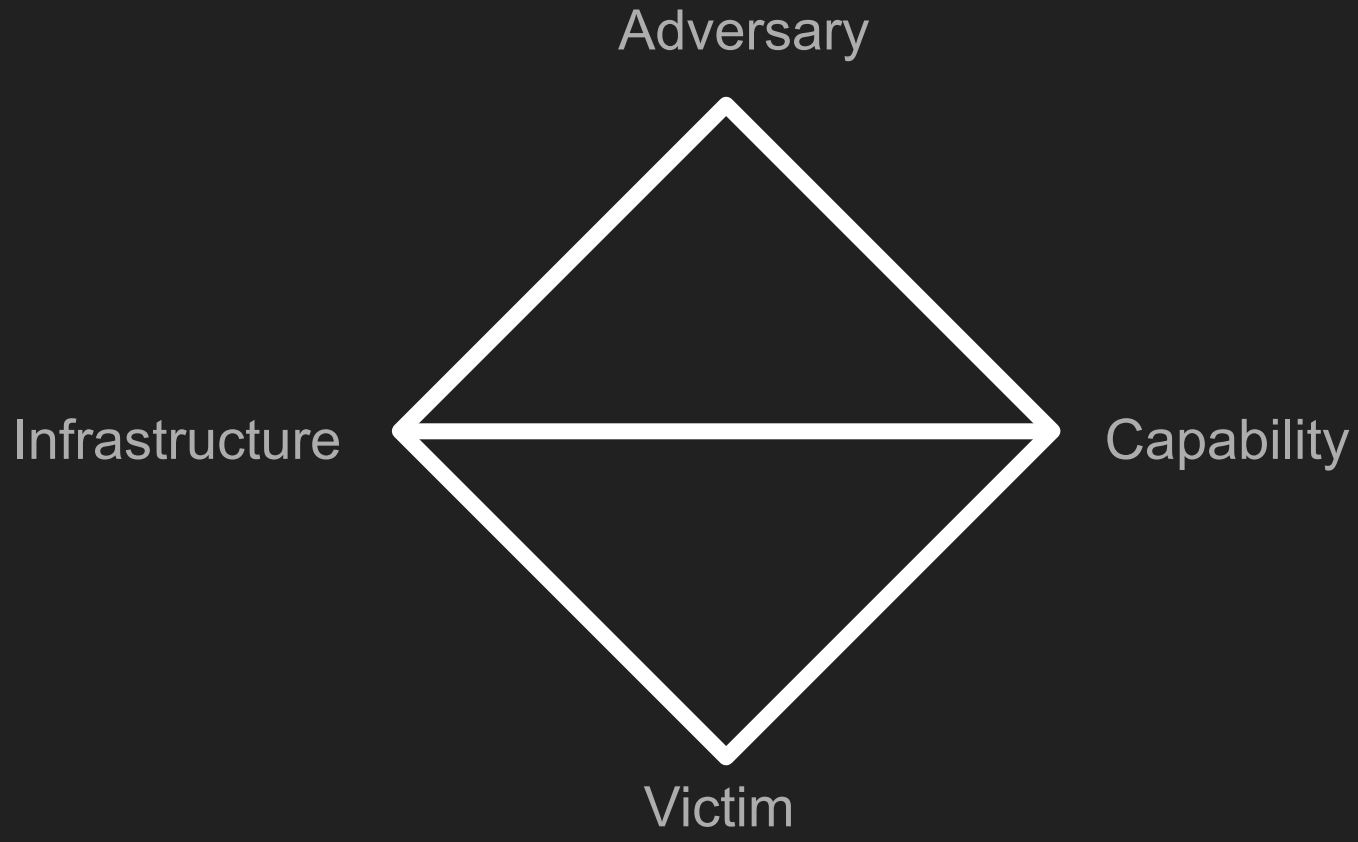
What are the laws they break?

US federal Computer Fraud and Abuse Act which prohibits:

- Accessing a protected computer
- Trafficking in malware
- Trafficking in passwords
- Accessing a government computer
- Spying using a computer
- Damaging a computer system

Investigated by the FBI in cases of counterintelligence, and the Secret Service in other cases

Diamond Model for Intrusion Analysis Caltagirone et al. 2013



Caltagirone's Axioms of the Diamond Model

1. **Adversaries** harm **victims** using **capabilities** on their **infrastructure**
2. There are always adversaries who want to harm victims
3. Every system is a potential victim, and always has vulnerabilities
4. Every attack is at least a two-step process
5. Attacks need external resources
6. Attackers always have some kind of relationship to the victims, however distant
7. Some adversaries and victims are persistently at odds

Reverse engineers’ job: know the enemy

“If you know the enemy and know yourself, you need not fear the result of a hundred battles. If you know yourself but not the enemy, for every victory gained you will also suffer a defeat.

If you know neither the enemy nor yourself, you will succumb in every battle.”

Sun Tzu, ~400 BC

“Know thine enemy” - Sun Tzu

Potentially Unwanted Programs (PUPs)

Programs of dubious value, which may be annoying, but are not seriously malicious.

There is a chance that someone out there values this software

“Know thine enemy” - Sun Tzu

Backdoor:

A program listening to the network for an attacker to remotely control the victim computer

“Know thine enemy” - Sun Tzu

Remote Access Trojan (RAT)

A subclass of backdoors which targets personal computers and often is capable of spying on or interacting with the user using the camera, microphone, and speakers. Creepy. A reason people cover their webcams.

“Know thine enemy” - Sun Tzu

Dropper/Downloader:

A program designed to download malware.

“Know thine enemy” - Sun Tzu

Cryptor:

A program designed to decrypt malicious code that has been disguised as nonsense using cryptography.

“Know thine enemy” - Sun Tzu

Keylogger or Info theft:

A program that steals information or user interactions.

“Know thine enemy” - Sun Tzu

Botnet:

A collection of machines with backdoors which can be controlled together to perform coordinated attacks (like DDoS)

“Know thine enemy” - Sun Tzu

Launcher:

A program that stealthily launches other programs, making it harder to detect that they are running. They may use tricks such as process injection to hijack an existing benign process and add a thread running malware code

“Know thine enemy” - Sun Tzu

Rootkit:

A malicious replacement or modification of a benign program, such as your operating system. Because antivirus programs are run by the operating system, a rootkit can hide from them and pretend it doesn’t exist.

“Know thine enemy” - Sun Tzu

Spam email sender

A program that sends out spam emails. IP addresses that send too many spam emails eventually get blocked, so attackers would much rather use other people’s IP addresses until they get banned.

MITRE ATT&CK Framework

Adversarial Tactics, Techniques, & Common Knowledge

- Variations for:
  - Enterprise networks
  - Mobile Devices
  - Clouds
  - Industrial Control Systems

Gives a framework for classifying attacks, comparing them to each other, and focusing on the most important information.

MITRE ATT&CK

1. Reconnaissance

MITRE ATT&CK

2. Resource Development

MITRE ATT&CK

3. Initial Access

MITRE ATT&CK

4. Execution

MITRE ATT&CK

5. Persistence

MITRE ATT&CK

6. Privilege Escalation

MITRE ATT&CK

7. Defensive Evasion

MITRE ATT&CK

8. Credential Access

MITRE ATT&CK

9. Discovery

MITRE ATT&CK

10. Lateral Movement

MITRE ATT&CK

11. Collection

MITRE ATT&CK

12. Command and Control

MITRE ATT&CK

13. Exfiltration

MITRE ATT&CK

14. Impact

Threat Intelligence, Malware Analyst, Vulnerability Analyst

(links will expire)

Sandia

NSA

CrowdStrike

John’s Hopkins APL

Expect >$70K to start, >$150K Mid-career

Class Structure

- Weekly assignments due Monday (First one due 1/29)
- Readings to be done by Tuesday
- ~20 minutes of lecture per class
- Balance of the time on practice/help on the assignment

Grades:
60% Projects
30% Midterm/Final
10% Participation/Attendance

Do you prefer quizzes on Tuesdays, or a big midterm/final?
If you choose quizzes, final exam will be a take-home project
Final Project 10%, 30% in-class quizzes, 2 lowest dropped.

Projects as a Portfolio

You will submit your assignments on Github. You can make a private repo and add me (jr-nm). BUT, I encourage you to make a public repo that you can show off in interviews, etc.

Never commit malware samples. Rather, reference them by their hash (We'll learn this soon)