Threat Hunting Professional

v2

Threat Hunting Hypothesis

Section 01 | Module 04

© Caendra Inc. 2020
All Rights Reserved

Table of Contents

Learning Objectives

In this module, we will discuss a threat hunting methodology that you can follow to start hunting in your environments.

MITRE ATT&CK

4.1 MITRE ATT&CK

One of the points we outlined in a previous module was that a good threat hunter must have knowledge of various types of attacks and often a deep understanding of how they manifest, which helps with knowing how and where to detect them.

Knowing about attacks also means knowing where to search for them. In many cases, it will be required that you identify how to detect the said attack, which requires you to simulate it and to fill knowledge gaps otherwise not provided by your sources.

4.1 MITRE ATT&CK

The detection methods can be based on specific artifacts left on the Operating System or in the Network traffic generated, which you can use to figure out where detection points.

The sources for attacks can be comprehensive, such as the [MITRE ATT&CK](https://attack.mitre.org/), vendor reports, white papers, conference talks, individual researchers, your organization's offensive team, etc. In this chapter, we will focus on the MITRE ATT&CK.

https://attack.mitre.org/

4.1 MITRE ATT&CK

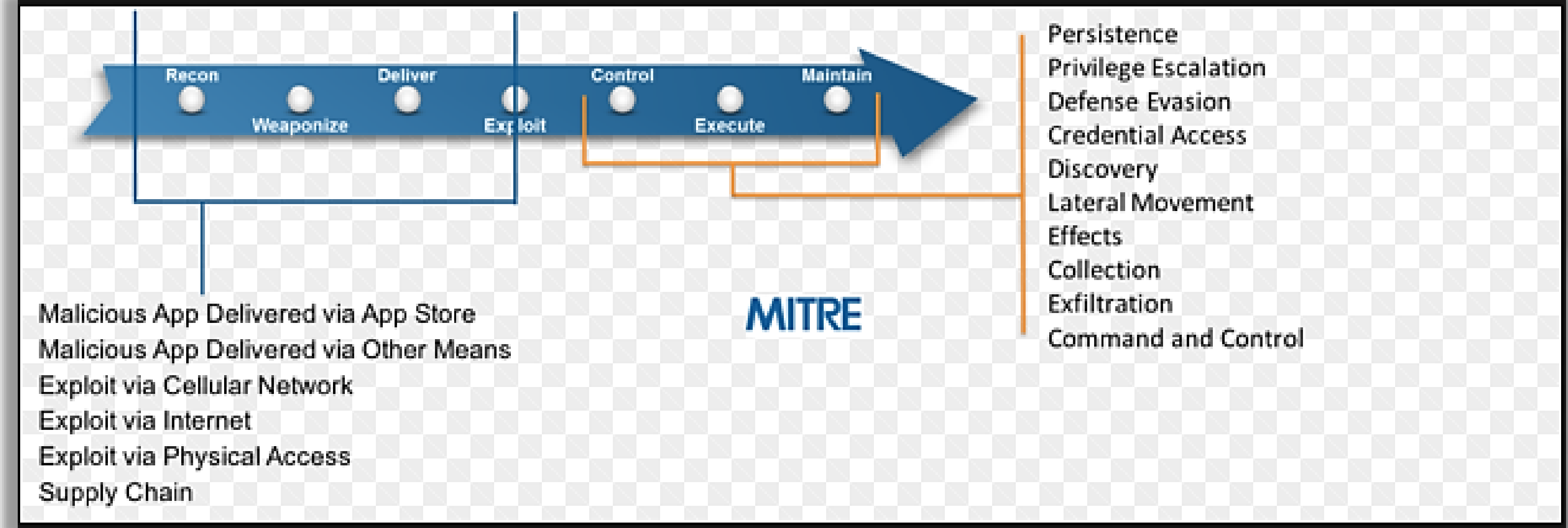
According to MITRE's definition, MITRE's Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) is a curated knowledge base and model for cyber adversary behavior, reflecting the various phases of an adversary's attack lifecycle and the platforms they are known to target.

On the next slide, you'll see a visual of MITRE's representation of the attack lifecycle and where ATT&CK fits.

https://www.mitre.org/sites/default/files/publications/pr-18-0944-11-mitre-attack-design-and-philosophy.pdf

4.1 MITRE ATT&CK

MITRE ATT&CK – Post Exploitation



4.1 MITRE ATT&CK

In the matrix, columns define tactics, while each box represents a technique. There are over 200 techniques.

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Commonly Used Port	Automated Exfiltration	Account Access Removal
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	Binary Padding	Bash History	Application Window Discovery	Application Deployment Software	Automated Collection	Communication Through Removable Media	Data Compressed	Data Destruction
External Remote Services	Command-Line Interface	Account Manipulation	AppCert DLLs	BITS Jobs	Brute Force	Browser Bookmark Discovery	Component Object Model and Distributed COM	Clipboard Data	Connection Proxy	Data Encrypted	Data Encrypted for Impact

https://attack.mitre.org/

4.1 MITRE ATT&CK

Each technique contains an explanation, procedure examples (often linked to threat reports), mitigation, and detection. It also includes metadata such as System requirements and Permissions Required to perform the technique.

Let's use "Kerberoasting" as an example.

4.1 MITRE ATT&CK

Kerberoasting

Service principal names (SPNs) are used to uniquely identify each instance of a Windows service. To enable authentication, Kerberos requires that SPNs be associated with at least one service logon account (an account specifically tasked with running a service [1]). [2] [3] [4] [5]

Adversaries possessing a valid Kerberos ticket-granting ticket (TGT) may request one or more Kerberos ticket-granting service (TGS) service tickets for any SPN from a domain controller (DC). [6] [7] Portions of these tickets may be encrypted with the RC4 algorithm, meaning the Kerberos 5 TGS-REP etype 23 hash of the service account associated with the SPN is used as the private key and is thus vulnerable to offline Brute Force attacks that may expose plaintext credentials. [7] [6] [5]

This same attack could be executed using service tickets captured from network traffic. [7]

Cracked hashes may enable Persistence, Privilege Escalation, and Lateral Movement via access to Valid Accounts. [4]

ID: T1208

Tactic: Credential Access

Platform: Windows

System Requirements: Valid domain account or the ability to sniff traffic within a domain.

Permissions Required: User

Data Sources: Windows event logs

Contributors: Praetorian

Version: 1.0

Created: 18 April 2018

Last Modified: 18 July 2019

https://attack.MITRE.org/techniques/T1208/

4.1 MITRE ATT&CK

Under the Detection section, we see a suggestion on how to detect this attack:

Detection

Enable Audit Kerberos Service Ticket Operations to log Kerberos TGS service ticket requests. Particularly investigate irregular patterns of activity (ex: accounts making numerous requests, Event ID 4769, within a small time frame, especially if they also request RC4 encryption [Type 0x17]). [1] [7]

https://attack.MITRE.org/techniques/T1208/

4.1 MITRE ATT&CK

Because ATT&CK has the most public attack techniques, we can use it to learn a lot about them. ATT&CK does not define only a single signature on how to detect certain techniques, and that is also good news. We want to detect any occurrence of the technique; we want to detect based on the adversary's TTPs.

Here is a video from BSides Bristol on practically utilizing ATT&CK for hunting.

https://www.youtube.com/watch?v=tmW60vC0tHE

4.2

Data Collection and Analysis

4.2 Data Collection and Analysis

Now that we are familiar with and can obtain information about various attacks, we need to be able to apply that knowledge against our data.

To start hunting, we need to determine what we want to hunt for, based on a hypothesis, and then perform the hunt by looking at the data we have.

4.2 Data Collection and Analysis

But what is the data that we have?

How do we ensure it is qualified to our needs?

How do we transform and make it useful to us?

4.2 Data Collection and Analysis

Before we hunt, we need to collect data. When collecting, we should ensure that we have a purpose based on what we want to find in that data to avoid collecting a mountain of ‘noise populated’ logs.

As mentioned in a previous module, the primary data types are host and network data.

4.2 Data Collection and Analysis

Once we identify what data we exactly want to collect, we need to find a method for exporting that data from the local machines to our analytics software.

Generally, the methods are:

- Push – agent on the host forwards log data as its captured
- Pull – remotely connect and collect data at the time of the connection
- A mix of the above

4.2 Data Collection and Analysis

With the exporting method in place, we should be able to conclude our hunting capability and expectations as a bare minimum based on the following:

- How much of the needed data is available for a hunt?
- How much of my environment would I cover with the available data during a hunt?
- How far back in time can I search?
- What about the quality of that data; for example, how consistent is it across the different data sources? How do we govern that?

4.2.1 Data Governance

[Data governance](https://searchdatamanagement.techtarget.com/definition/data-governance) (DG) is the overall management of the availability, usability, integrity, and security of data used in an enterprise. Businesses benefit from data governance because it ensures data is consistent and trustworthy.

https://searchdatamanagement.techtarget.com/definition/data-governance
https://bi-survey.com/wp-content/uploads/2017/11/Data-Governance-topics.png

4.2.1 Data Governance

With DG, we can define data quality that ensures, among other things:

- **Data completeness** - Is all required data available and for how long?
- **Data consistency** - Is a standard naming convention applied to data fields from different data sources?
- **Data timeliness** – Do timestamps represent the actual creation time of the events?

4.2.1 Data Governance

The benefit of Data Governance should be clear to us. So, how do we identify anomalies in that data?

A good first step is to understand what is “normal” in the environment, because if you do not know what is currently there, it is much harder to identify what should not be.

4.2.1 Data Governance

Getting familiar with "normal" can be achieved by the baselining of regular activity that is expected on a golden image of an endpoint and in the network traffic. This includes:

- Running Processes
- User logons -> where, when, and what type of login
- Network connections
- Services and scheduled tasks
- Software that is allowed to execute

4.2.2 Data Analysis

And that brings us finally to data analysis, that is how do we get the useful parts of the collected data?

An analysis is usually performed on a SIEM system, such as ELK/HELK, Splunk, Graylog, and many others.

4.2.2 Data Analysis

Analyzing the data means that we are manipulating it by:

- Searching it
- Aggregating it
- Filtering it
- Joining data together

What is searching and aggregating?

4.2.2 Data Analysis

Searching data gives us the ability to find answers to our questions. In practice, the first searches we perform will most likely not meet our expectations. It will have to be modified in one way or another (e.g., reduce the timespan or exclude a user, etc.) for us to obtain what we want.

4.2.2 Data Analysis

Other searches may result in more questions rather than answers. Searches (and the overall detection) will nearly always be a work in progress. We should ensure that gaps and assumptions are clear.

Overall, the big benefit is that searching will present us with the ability to identify anomalies.

4.2.2 Data Analysis

The searching queries will be specific to the tool that you are using for analysis, but in general, the tool shouldn't matter, as their syntax is similar. Most (if not all) support Boolean operators, comparison operators, direct key-value comparison, wildcard matching, and many more.

The more searches you do, the better you become at it. Practice is crucial!

4.2.2 Data Analysis

Aggregation, on the other hand, is grouping values within the same field together. In many cases, aggregated data is sorted by either the amount of the most common or least common data in that field – it gives us a different perspective in the data. We can see an oversimplified example in the table below:

Process name	Occurrences
Svchost.exe	13
Winword.exe	9
Calc.exe	1

4.2.2 Data Analysis

The aggregated data and terms displayed can be extended to match your exact criteria. For example, if you are looking at executed PowerShell commands, you may want to display:

- The command itself
- Who executed it
- On which host was it executed
- How many times it was executed

4.2.2 Data Analysis

In many occasions, we will do a search, which will be aggregated afterward. One example is looking at the least common occurrence of a search, which returned a lot of results. By looking at the aggregated data, we attempt to identify an outlier/odd occurrences.

Naturally, through aggregation, we are performing statistical analysis (count, sum, average of, frequency analysis, etc.) on the data, which gives us a different perspective into it that can outline the abnormal.

4.2.2 Data Analysis

As hunters, you should expect that some of the results may contain only partial answers on what you are looking for. So how do you continue?

Remember how we discussed multiple data sources? The answer is by switching to other data sources (when possible).

4.2.2 Data Analysis

A great example of this is an anomaly that someone attempted a connection to a suspicious IP address.

You can correlate this with PCAP/NetFlow data to review exactly what the communication contained and to verify if it is a true malicious event, or disregard it as a false positive.

4.3

Hunting Hypothesis and Methodology

4.3 Hunting Hypothesis and Methodology

Every hunt begins by defining a hypothesis. It consists of the following:

- Identify the specific behavior we want to hunt for
- Understand the attack technique behind it
- Identify what data (and from which source) we need to detect it

To achieve this, we'll use a 5-step process.

4.3 Hunting Hypothesis and Methodology

The 5 steps are:

1. Pick a tactic and technique
2. Identify associated procedure(s)
3. Perform an attack simulation
4. Identify evidence to collect
5. Set scope

4.3.1 Pick a Tactic and Technique

Step #1 Pick a tactic and technique

In this step, we review attack sources and pick a technique. As mentioned earlier, we will utilize the MITRE ATT&CK framework in this course, so that is our source.

As an example, let's pick technique T1502 - Parent PID Spoofing under "Privilege Escalation".

https://attack.mitre.org/techniques/T1502/

4.3.1 Pick a Tactic and Technique

Parent PID Spoofing:

"Adversaries may spoof the parent process identifier (PPID) of a new process to evade process-monitoring defenses or to elevate privileges. … Adversaries may abuse these mechanisms to evade defenses, such as those blocking processes spawning directly from Office documents and analysis targeting unusual / potentially malicious parent-child process relationships … Explicitly assigning the PPID may also enable Privilege Escalation …"

https://attack.mitre.org/techniques/T1502/

4.3.2 Identify Associated Procedure(s)

Step #2 Identify associated procedure(s)

ATT&CK will also present us with the procedure for the technique. In our case, it has the following:

Procedure Examples

Name	Description
Cobalt Strike	Cobalt Strike can spawn processes with alternate PPIDs.[6]

4.3.2 Identify Associated Procedure(s)

We recommend additional research in this phase, as reports and blog posts often have developed tools and different implementations of the technique, which may not be present in ATT&CK. Do your own research to really understand the procedures, including their prerequisites, requirements, and outcome if successfully executed.

The following posts provide additional context to our example of PPID Spoofing, which you can check out here and here.

https://medium.com/@r3n_hat/parent-pid-spoofing-b0b17317168e
https://blog.f-secure.com/detecting-parent-pid-spoofing/

4.3.2 Identify Associated Procedure(s)

From the blog posts, we can tell:

- **Prerequisite** – the attacker must have already compromised the box and can execute commands.
- **Requirements** – process path, path to injected assembly, and fake parent process name.
- **Outcome** – new processes are started.

4.3.3 Perform an Attack Simulation

Step #3 Perform an attack simulation

Although not always necessary (based on what you have gathered previously), this step replicates the procedure(s) of the technique. During the replication process, you can observe what data and logs are generated, and in the next step, build your detection based on the identified behaviors.

It can be very time consuming, as it requires you to have the systems and monitoring capabilities equivalent to the ones in your environment.

4.3.4 Identify Evidence to Collect

Step #4 Identify evidence to collect

Once executed in our controlled environment, we should start investigating places that may contain artifacts of interest; this could be on disk, in-memory, network traffic, registry, etc.

The goal of this step is to identify behaviors we can use to detect this activity in the future.

4.3.4 Identify Evidence to Collect

Some of the things to look for are:

- Deviations from baselines
- Attempts to appear normal (e.g., svchost.exe running from user directory)
- Unexpected encryption
- Odd frequency of occurrence (e.g., too many connections or user activity at an abnormal time)

4.3.4 Identify Evidence to Collect

Other than identifying detection behaviors, you will likely find some false positives, which may be unique to your environment. Having the ability to filter these out at an early stage is beneficial during the actual hunt.

To avoid confusion at this point (due to concepts not yet explained), we will return to the PPID Spoofing detection at a later stage during a lab.

4.3.5 Set Scope

Step #5 Set Scope

Finally, now that we know everything about the technique and its procedure(s) and having identified behaviors after the simulation, we are almost ready to begin proving/disproving our hypothesis of whether a specific activity has occurred or not.

The last thing left is to define the hunting scope.

4.3.5 Set Scope

The main things we should define in the scope are:

- Hunt Duration
- What data and where to collect from

We recommend that the duration of a hunt is at least a week.

4.3.5 Set Scope

As for the scope, there may be other external factors that do not allow us to capture all data from all devices. This could be due to network bandwidth or analysis capability in the defined timeframe (too many logs to analyze).

In some hunts, you may want to focus only on Business-Critical Systems, while on others only on certain data sources.

4.3.5 Set Scope

You should outline the assumptions and limitations set in the scope, which may have an impact on future decisions (about tools, architecture, future hunts, etc.).

For our example, of PPID spoofing, we define it as:

- Time – 1 week
- Collection – All Windows devices sending Event Tracing for Windows (ETW) logs. We will discuss ETW in a later module; for now, think of it as a certain type of Windows logs.

4.3.5 Set Scope

Remember to take notes during the hunts of any activity that was unsolved, as these may be used at a later stage or during another hunt.

Use a tool that you are comfortable with, and you will want to ensure a consistent structure to help you identify notes from previous hunts.

Hunting Metrics

4.4 Hunting Metrics

Hunting metrics are a way for us to track our progress and report back to management about our hunting expeditions.

But how do we define metrics that actually make sense? Are we good at hunting only if we find malicious activity? No, simply because there may be nothing to find.

4.4 Hunting Metrics

You may be thinking that ongoing simulated malicious activity may be good to show that your hunting expeditions can detect that activity, but simulating malicious activity in production environments is not advised.

Instead, coordination with the penetration testing team can be formed to test the detection capabilities.

4.4 Hunting Metrics

Some things that we as hunters have control over are great to show the work we perform, such as:

- How often we hunt
- Technique coverage of ATT&CK
- Procedure coverage of certain groups relevant for your type of business
- Hunting coverage of the network
- Historic logging capability that facilitates our hunts (how far back can we go)

References

References

Report Template for Threat Intelligence and Incident Response

https://zeltser.com/cyber-threat-intel-and-ir-report-template/

MITRE ATT&CK

https://attack.mitre.org/

MITRE ATT&CK™: Design and Philosophy

https://www.mitre.org/sites/default/files/publications/pr-18-0944-11-mitre-attack-design-and-philosophy.pdf

MITRE ATT&CK Framework For Threat Hunting

https://www.youtube.com/watch?v=tmW60vC0tHE

References

[What is data governance and why does it matter?](https://searchdatamanagement.techtarget.com/definition/data-governance)

https://searchdatamanagement.techtarget.com/definition/data-governance

[Parent PID Spoofing](https://attack.mitre.org/techniques/T1502/)

https://attack.mitre.org/techniques/T1502/

[Parent PID Spoofing](https://medium.com/@r3n_hat/parent-pid-spoofing-b0b17317168e)

https://medium.com/@r3n_hat/parent-pid-spoofing-b0b17317168e

[Detecting Parent PID Spoofing](https://blog.f-secure.com/detecting-parent-pid-spoofing/)

https://blog.f-secure.com/detecting-parent-pid-spoofing/

References

[T1208 - Kerberoasting](https://attack.MITRE.org/techniques/T1208/)

https://attack.MITRE.org/techniques/T1208/