MITRE Mapping Playbook

# Purpose

The MITRE Mapping Playbook provides defensive recommendations for all Sentinel Alerts.

It does this by mapping the hard-coded MITRE Technique values in each Sentinel rule to the MITRE SHIELD information that is stored in 2 watchlist reference tables.

The resulting information is added to the Sentinel incident’s comments section.

**Future Thoughts**

This playbook is structured so that it can be extended to include auto-execution of additional playbooks in the future.

# High Level Operation

Diagram

Description automatically generated

**What are the MITRE SHIELD categories:**

DTE - Defender Technique - Primary Defense Subject

DOC - Defender Opportunities - Conditions where this defense method applies.

DUC - Defender Use Cases - Examples where this defense method applies.

DPR - Defender Response Procedures - Actions taken to respond.

# Implementation Summary

The following steps are taken to install/configure the Mitre Mapping Playbook

* Import the playbook
* Import the attack and mitre\_shield watchlist tables
* Configure automation so that the playbook triggers on every new incident

(Step by step instructions are out of scope for this document)

Note: It is recommended to maintain backup copies of the playbook and the watchlist tables.

# Introducing the ‘attack’ and ‘mitre\_shield’ Tables

There are 2 tables used by the MITRE Mapping playbook: ‘attack’ and ‘mitre\_shield’:

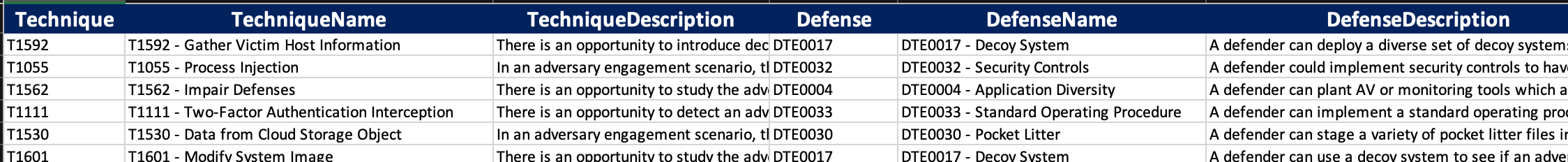
‘attack’ Table

Maps all the MITRE ATT&CK techniques to the MITRE SHIELD DTE values.

Note: It’s unlikely this table will need to be customized.

KQL to view attack table:

\_GetWatchlist('attack')|project Technique, TechniqueName, TechniqueDescription, Defense, DefenseName, DefenseDescription



## ‘mitre\_shield’ Table

Maps DTE to the associated DUC, DOC and DPR values.

**Note: It’s likely this table WILL need to be customized to suit an organization’s specific defensive practices (see next section for a demonstration).**

KQL to view mitre\_shield table:

\_GetWatchlist('mitre\_shield')|project DTE,DTEDescription,DUC,DUCDescription, DOS,DOSDescription,DPR,DPRDescription

Table

Description automatically generated

# Demo: Customizing the mitre\_shield table

MITRE SHIELD provides a SOC with a reference of defensive information intended to help make decisions on how to defend against specific adversarial techniques as defined by ATT&CK.

The primary topics of interest in SHIELD are:

DTE – Defensive Technique. There’s around 30 of these.

Underneath the DTE are 3 subcategories:

DUC – Defensive Use Case

DOS – Defensive Opportunity

DPR – Defensive Response Procedure

DUC Tip: You may consider hiding/igoring DUC if the SOC operator isn’t interested in these general categories. Many of SHIELD’s DUC references relate to setting up ‘decoys’. If this is out of scope, then focus should be placed on mapping to the remaining SHIELD defensive categories. See Appendix A for more details.

DOS Tip: You may consider hiding/ignoring DOS unless you’re interested in understanding the vulnerabilities associated with an attacker’s actions – eg. Once an attacker has established C2, what dependencies help him maintain that control? Understanding an attacker’s dependencies are ‘opportunities’ to control the attacker’s actions.

DPR Tip: Focus on this as the place to customize organizational specific response procedures.

## Example: Analytics rule Password Spray – T1110

Step 1: Find the DTE that maps to the Tactic number in the analytic rule.

Open the ‘attack’ watchlist and find the DTEs that map to T1110 (there could be 1 or more)



Step 2: Find and edit the DPRs associated with the DTE

Now open the ‘mitre\_shield’ watchlist and edit the DPR entries that map to the above DTEs



Alternatively, add new lines to provide additional response procedure recommendations. Eg:

* Block the source IP
* Lock out the username
* Isolate the machine

The full list of DTEs in Appendix A, and DPRs is listed in Appendix B. Feel free to add custom DTEs and DPRs.

Note: Multiple DTEs mapped to a single Tactic is supported. And Multiple DPRs mapped to a single DTE is also supported.

Note: There is a limit of 3,000 characters per field.

# Appendix A: MITRE SHIELD Tactics

For more information on any of the descriptions below, go to <https://engage.mitre.org/> and enter the description into the search box in the top right corner of the page.

Some links have already been added below for some of the common points of interest.

Tip: **Blue** represents defensive techniques and **Red** represents decoy/deception techniques.

|  |  |
| --- | --- |
| **DTE** | **DTEDescription** |
| DTE0001 | DTE0001 - Admin Access |
| DTE0003 | DTE0003 - API Monitoring |
| DTE0004 | [DTE0004 - Application Diversity](https://engage.mitre.org/activities/EAC0006/) |
| DTE0005 | DTE0005 - Backup and Recovery |
| DTE0006 | [DTE0006 – Baseline](https://engage.mitre.org/activities/EAC0019/) |
| DTE0007 | DTE0007 - Behavioral Analytics |
| DTE0008 | DTE0008 - Burn-In |
| DTE0010 | DTE0010 - Decoy Account |
| DTE0011 | DTE0011 - Decoy Content |
| DTE0012 | DTE0012 - Decoy Credentials |
| DTE0013 | DTE0013 - Decoy Diversity |
| DTE0014 | DTE0014 - Decoy Network |
| DTE0015 | DTE0015 - Decoy Persona |
| DTE0016 | DTE0016 - Decoy Process |
| DTE0017 | DTE0017 - Decoy System |
| DTE0018 | DTE0018 - Detonate Malware |
| DTE0019 | DTE0019 - Email Manipulation |
| DTE0020 | DTE0020 - Hardware Manipulation |
| DTE0021 | DTE0021 – Hunting |
| DTE0022 | DTE0022 – Isolation |
| DTE0023 | DTE0023 - Migrate Attack Vector |
| DTE0025 | DTE0025 - Network Diversity |
| DTE0026 | DTE0026 - Network Manipulation |
| DTE0027 | DTE0027 - Network Monitoring |
| DTE0028 | DTE0028 - PCAP Collection |
| DTE0029 | [DTE0029 - Peripheral Management](https://engage.mitre.org/activities/EAC0010/) |
| DTE0030 | [DTE0030 - Pocket Litter](https://engage.mitre.org/activities/EAC0011/) |
| DTE0031 | DTE0031 - Protocol Decoder |
| DTE0032 | DTE0032 - Security Controls |
| DTE0033 | DTE0033 - Standard Operating Procedure |
| DTE0034 | DTE0034 - System Activity Monitoring |
| DTE0035 | DTE0035 - User Training |
| DTE0036 | DTE0036 - Software Manipulation |

# Appendix B: MITRE SHIELD DPR List

|  |  |
| --- | --- |
| **DPR** | **DPR Description** |
| DPR0003 | Hook the Win32 Sleep() function so that it always performs a Sleep(1) instead of the intended duration. This can increase the speed at which dynamic analysis can be performed when a normal malicious file sleeps for long periods before attempting additional capabilities. |
| DPR0004 | Hook the Win32 NetUserChangePassword() and modify it such that the new password is different from the one provided. The data passed into the function is encrypted along with the modified new password, then logged so a defender can get alerted about the change as well as decrypt the new password for use. |
| DPR0005 | Trace activity through WinSock TCP API functions to view potentially malicious network events. Log it such that it can be pushed to a centralized location and analyzed further. |
| DPR0006 | Hook the Win32 DeleteFile() function to log all attempts at deleting a given file. This information can be used to trigger restoration attempts on critical data, reducing potential disruption if those files are unavailable for prolonged periods of time. |
| DPR0007 | Use a mix of vulnerable and nonvulnerable software on a system to allow you to see what exploits the adversary leverages in their attacks. |
| DPR0008 | Install Anti-virus or other end-point detection tools on systems to see if an adversary takes note of them and if so, how they react. |
| DPR0009 | Backup data on public facing websites and retain the files offline. In the event of data damage or loss, restore the data from backup. |
| DPR0010 | Backup data on an end-user system and store offline. If an adversary alters or deletes data on the system, restore the data using the backup copy. |
| DPR0011 | Maintain a verified baseline firewall configuration and use that copy as a fallback if an adversary alters that information. |
| DPR0012 | Maintain a verified list of group policies enforced on a system and use that copy if an adversary attempts to deviate from the baseline. |
| DPR0013 | Use behavioral analytics to detect Living Off The Land Binaries (LOLBins) being used to download and execute a file. |
| DPR0014 | Use behavioral analytics to identify a system running development tools, but is not used by someone who does development. |
| DPR0015 | Use behavioral analytics to identify abnormal system processes being used to launch a different process. |
| DPR0016 | Configure a decoy system and allow it to be used in an manner such that it collects activity logs and appears to be to be a legitimate system. |
| DPR0017 | Configure a system to generate internet browser traffic for a decoy user profile, creating artifacts such as cookies, history, temp files, etc. |
| DPR0018 | Alter the output of an adversary's profiling commands to make newly-built systems look like the operating system was installed months earlier. |
| DPR0019 | Alter the output of adversary recon commands to not show important assets, such as a file server containing sensitive data. |
| DPR0020 | Create a user account with a specified job function. Populate the user account's groups, description, logon hours, etc., with decoy data that looks normal in the environment. |
| DPR0021 | Create a user that has a valid email account. Use this account in such a way that the email address could be harvested by the adversary. This can be monitored to see if it is used in future attacks. |
| DPR0022 | Create directories and files with names and contents using key words that may be relevant to an adversary to see if they examine or exfiltrate the data. |
| DPR0023 | Seed a file system with content that is of no value to the company but reinforces the legitimacy of the system if viewed by an adversary. |
| DPR0024 | Create user credentials for a decoy account, such as 'User ABC'. Store those credentials in the browser and other places on the system to see if an adversary attempts to harvest them. |
| DPR0025 | Use a Windows Virtual Machine (VM) and a Mac VM to visit a malicious website and note any differences in how the site functions based on the client that was used. |
| DPR0026 | Deploy multiple decoy systems, each with a unique network fingerprint (ports, services, connections, etc.) in order to provide an adversary a wide range of targets. |
| DPR0027 | Create an isolated network populated with decoy systems that can be used to study an adversary's tactics, techniques, and procedures (TTPs). |
| DPR0028 | Use a segregated network to visit a compromised site. If the machine becomes infected, allow the machine to remain on with internet access to see if an adversary engages and takes action on the system. |
| DPR0029 | Create a persona that represents an employee with hobbies, outside interests, personal accounts, etc. This persona may be used in conjunction with decoy accounts and credentials. |
| DPR0030 | Create a persona that represents an employee's projects and job scope. This persona information can be leveraged in conjunction with Burn-In and Pocket Litter. |
| DPR0031 | Create decoy processes on a system that mimic common antivirus process names. These processes when seen may prevent adversary malware from executing for fear of detection. |
| DPR0032 | Use an isolated system to visit a suspected compromised website. Collect any associated scripting code or files dropped onto the system. |
| DPR0033 | Setup a server which appears to be something that is commonly expected within a network, such as web server. |
| DPR0034 | Take malware received via spearphishing and detonate it on an isolated system in order to collect execution and network communication artifacts. |
| DPR0035 | Detonate a malware sample in a decoy network to engage with an adversary and study their TTPs. |
| DPR0036 | Modify the destination of inbound email to facilitate the collection of inbound spearphishing messages. |
| DPR0037 | Remove the microphone from a laptop to prevent an adversary from capturing audio from the device. |
| DPR0038 | Remove the Wi-Fi hardware from a device to prevent an adversary from enabling and using a Wi-Fi connection. |
| DPR0039 | Pivot on Command and Control information to identify other infrastructure used by the same adversary. |
| DPR0040 | Unplug an infected system from the network and disable any other means of communication. |
| DPR0041 | When malware is received via spearphishing, move the email message onto a decoy system prior to detonating the malicious file attachment. |
| DPR0043 | Deploy a mix of network devices (systems, servers, printers, phones, etc.) to make a decoy network look realistic. |
| DPR0044 | Deploy a variety of systems which reflect the use of multiple operating systems, hardware platforms, network services, etc. |
| DPR0045 | Add a kill switch to a decoy network that can be used to shutdown all network communication if an adversary takes an action that is out of the desired scope. |
| DPR0046 | Introduce intermittent network packet loss on a decoy network to interfere with an adversary's activities. |
| DPR0047 | Capture network logs for internet-facing devices and send those logs to a central collection location. |
| DPR0048 | Capture all network device (router, switches, proxy, etc.) logs on a decoy network and send those logs to a central collection location. |
| DPR0049 | Collect PCAP on a decoy network to improve visibility into an adversary's network activity. |
| DPR0050 | Introduce external devices (e.g. a USB drive) to a machine in an adversary engagement scenario to see how quickly an adversary gains awareness to its presence and if they attempt to leverage the device. |
| DPR0051 | Configure controls (such as AutoRun) which would require an adversary to take additional steps when leveraging a peripheral device to execute their tools. |
| DPR0052 | When staging a decoy system and user account, populate a user's folders and web history to make it look realistic to an adversary. |
| DPR0053 | Stage a USB device with documents on a specific topic in order to see if they are exfiltrated by an adversary. |
| DPR0054 | Create and apply a decoder which allows you to view encrypted and/or encoded network traffic in a human-readable format. |
| DPR0055 | Weaken security controls on a system to allow for leaking of credentials via network connection poisoning. |
| DPR0056 | Implement policies on a system to prevent the insecure storage of passwords in the registry. This may force an adversary to revert these changes or find another way to access cached credentials. |
| DPR0057 | Require approvals and waivers for users to make changes to their system which requires administrative access. Any changes not made through this process are suspect and immediately investigated as malicious activity. |
| DPR0058 | Create a development library that all users must leverage in order to interact with any hosted databases. This library modifies queries to look difficult to write. Any queries made without the library will now be obvious to detect and are immediately investigated as malicious activity. |
| DPR0059 | Ensure that systems capture and retain common system level activity artifacts that might be produced. |
| DPR0060 | Monitor Windows systems for event codes that reflect an adversary changing passwords, adding accounts to groups, etc. |
| DPR0061 | Train users to immediately report suspicious emails. Those emails could then be used for malware detonation or adversary engagement purposes. |
| DPR0062 | Train users to report potentially compromised devices so they can be isolated or migrated into deception networks. |
| DPR0063 | In an adversary engagement situation, if an adversary deletes or alters files on a machine they are controlling, restore the data to it original state and location to see how the adversary reacts. |
| DPR0064 | Modify the contents of an email message to maintain continuity when it is used for adversary engagement purposes. |
| DPR0065 | Use information about an adversary's TTPs to perform retroactive searches for any activity that have gone undetected. |
| DPR0066 | Run all user applications in isolated containers to prevent a compromise from expanding beyond the container's boundaries. |

# Appendix C: MITRE ATT&CK and SHIELD References

<https://shield.mitre.org>

<https://mitre-attack.github.io/attack-navigator/>

<https://github.com/mitre-attack/attack-datasources>

MITRE Tactics mapped to SHIELD

<https://engage.mitre.org/attack_mapping/>

Note: MITRE SHIELD is now MITRE Engage. This is a work in progress by MITRE.