SIEM:

Escape & Evade

Agenda

1. SIEM and its detection shortcomings!
   - Understanding SIEM deployment profiles
   - Agent interruption
   - Log source disruption
   - Rules bypass via subverting detection logic or command obfuscation
2. Advice for mitigations

$whoami

Dan

- Presales Director, UK at Vectra AI
- in/crossleydaniel/

Niall

- Professional Services
- MDR/IR
- Blue Team
- in/niallerrity/

Guy

- SecOps Veteran
- Business/People/Process SME

gkramer@ciauk.ltd
/guy-kramer/

Real-life Example

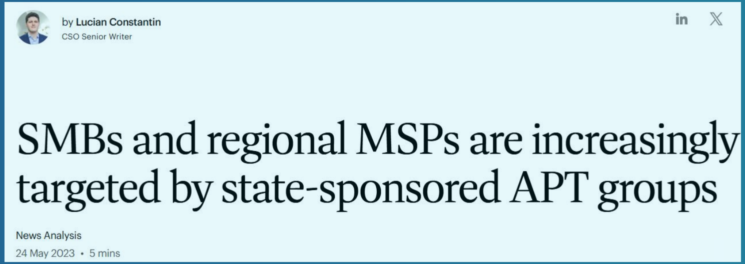
Actor: 2017: Operation CloudHopper

Desc: Avoided detection for approx 3yrs within MSPs (DXC, CGI, BAE… others), gaining access in to client networks *“the biggest corporate espionage efforts in history”*

Actor: APT10 Chinese Govt

Target: Intellectual property sensitive data

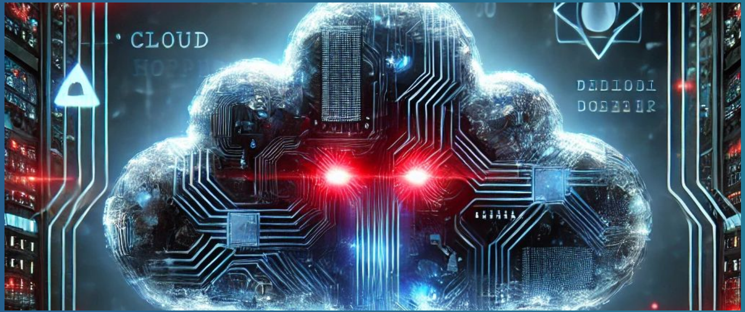
PwC/BAE report available here



by Lucian Constantin
CSO Senior Writer

SMBs and regional MSPs are increasingly targeted by state-sponsored APT groups

News Analysis
24 May 2023 • 5 mins



SIEM Core Concepts

What is a SIEM? & Example Deployment Profiles

- Security Information & Event Manager - it's where you shovel all your logs!
- Has been one of the core threat detection tools for the SOC for many decades.
- Deployments come in many different shapes and sizes. An understanding of the likely deployment profile will give you an understanding of the likely shortcomings..
    - E.g. Small/large deployments, outsourced Tier 1 etc..

Microsoft Sentinel

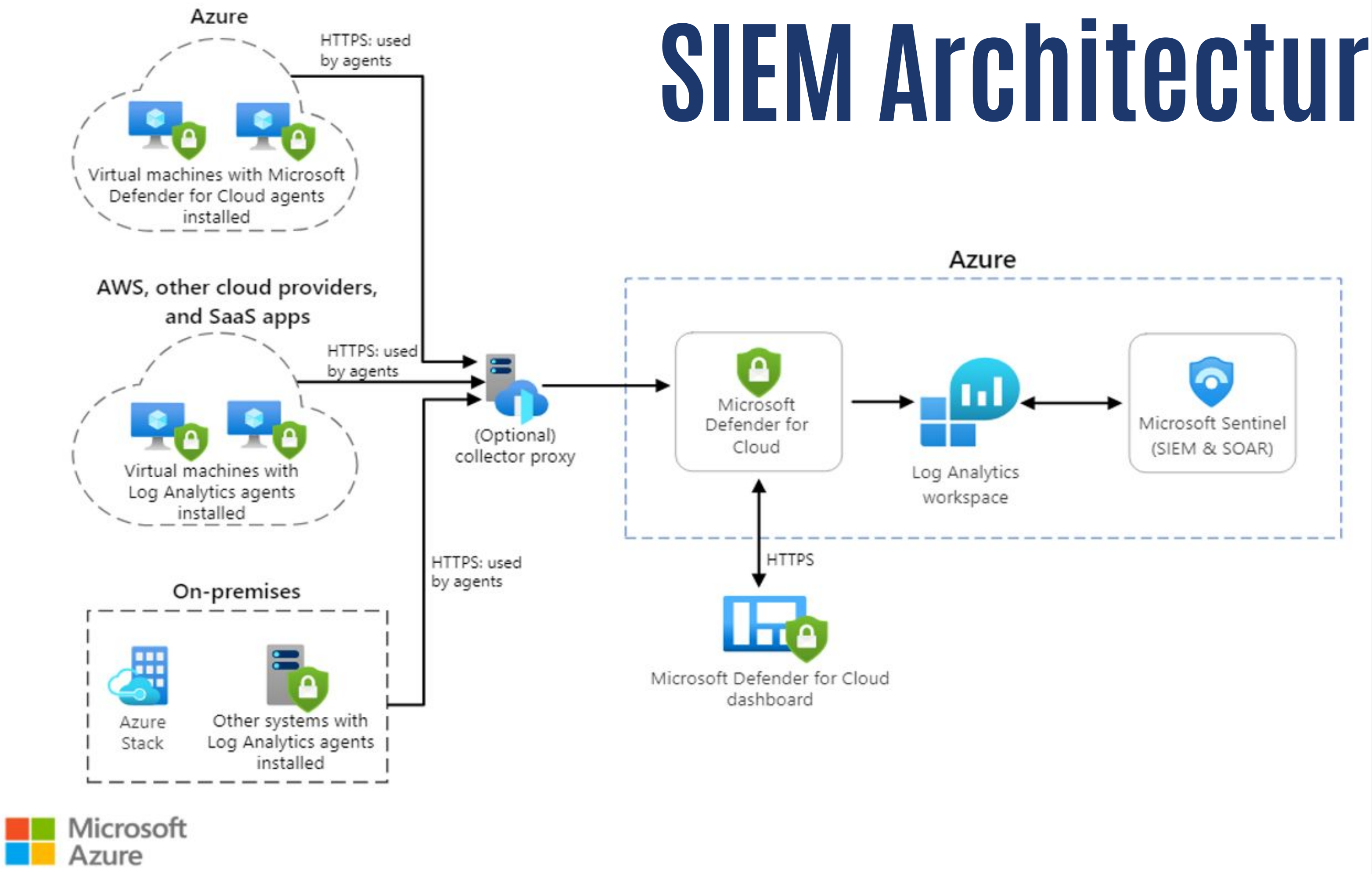
splunk>

Chronicle

Common SIEM Deployment Profiles

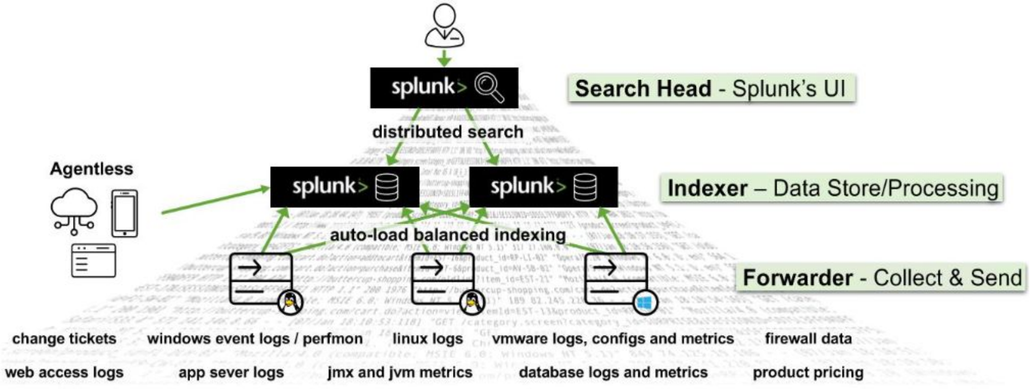
	Small Enterprise 1-1000 Employees	Mid Enterprise 1000-10000 Employees	Large Enterprise 10000+ Employees
Driving factors	Compliance	Compliance, Central Logging, SOC (Threat Intel & Advanced Analysis)	Compliance, Central Logging, SOC (Threat Intel & Advanced Analysis), Hunting
Managed	Outsourced/Minimal	Outsourced/Sometimes Specialists	Dedicated Team & Outsourced
Data Sources	Simple as network is usually not complicated. Firewalls, endpoints and maybe cloud	Hybrid with multiple security tools. Firewalls, NAC, Endpoint, Cloud Logs, IdP	Multiple environments with different tools, usually limited to a number of data sources due to cost.
Preferred SIEM Type	Cloud-based, managed service	Scalable, hybrid-capable	On-premises or hybrid with full customization
IOC Creation	No	Sometimes/Minimal	Both in-house and 3rd party
Typical Providers	Splunk Cloud or LogRhythm SaaS	IBM QRadar or Microsoft Sentinel	Splunk Enterprise or ArcSight
Weakness Analysis	Cloud based and lack of management	Lack of process or ownership around IOC management & investigations	Lack of data sources, correlation issues

SIEM Architecture


Azure
HTTPS: used by agents
Virtual machines with Microsoft Defender for Cloud agents installed
AWS, other cloud providers, and SaaS apps
HTTPS: used by agents
Virtual machines with Log Analytics agents installed
On-premises
Azure Stack
Other systems with Log Analytics agents installed
HTTPS: used by agents
(Optional) collector proxy
Azure
Microsoft Defender for Cloud
Log Analytics workspace
Microsoft Sentinel (SIEM & SOAR)
HTTPS
Microsoft Defender for Cloud dashboard
Microsoft Azure

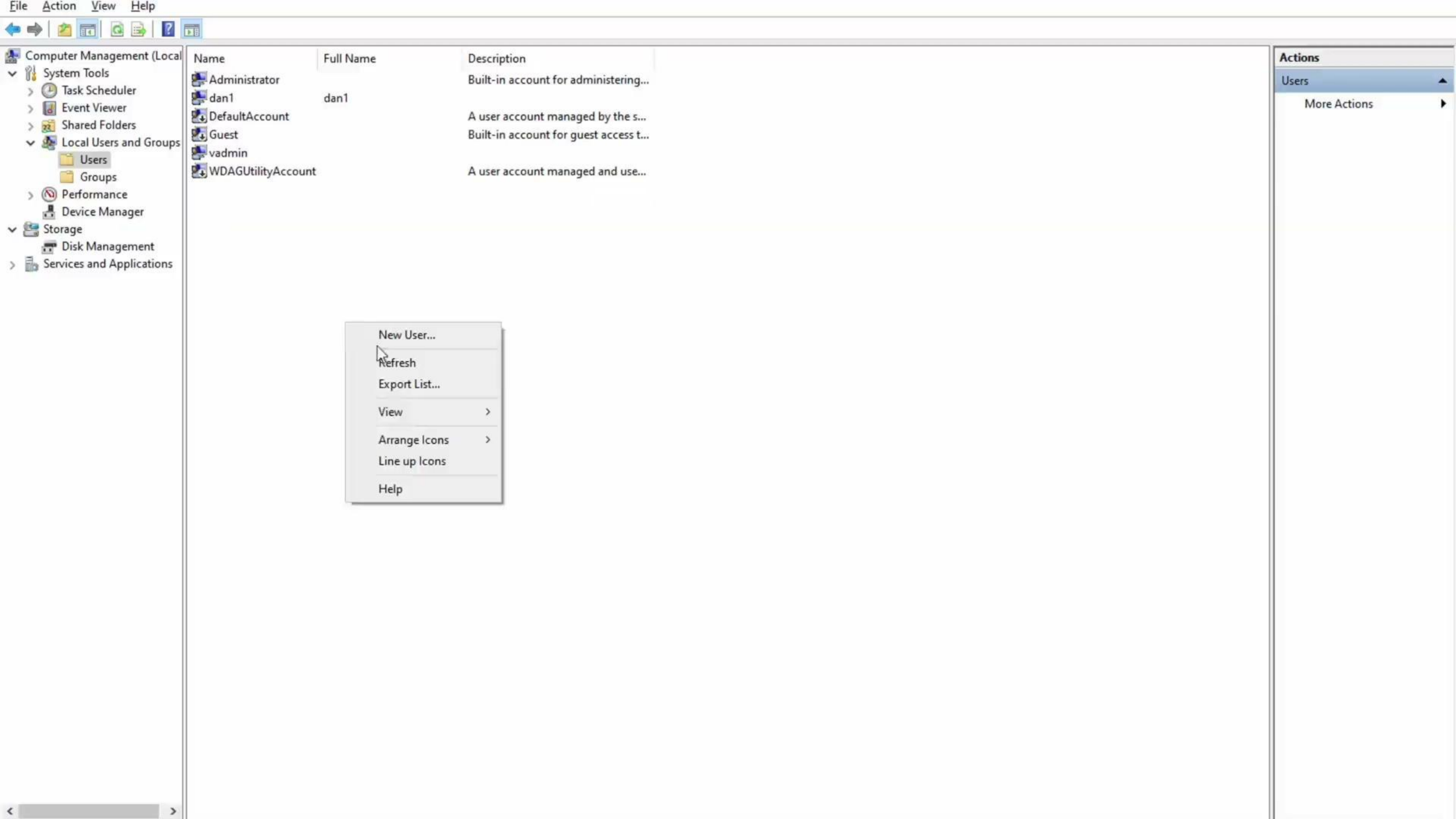
SIEM Log Sources

- For a SIEM to work, you need to get data into it!
  - Commonly via an agent, e.g. AMA
- Log source collection mechanisms is commonly via API, syslog, Windows event log, flat-file etc.
- Any issues with log collection can lead to missed detections!


splunk>
Search Head - Splunk's UI
distributed search
Agentless
splunk>
splunk>
Indexer – Data Store/Processing
auto-load balanced indexing
Forwarder - Collect & Send
change tickets
windows event logs / perfmon
linux logs
vmware logs, configs and metrics
firewall data
web access logs
app sever logs
jmx and jvm metrics
database logs and metrics
product pricing

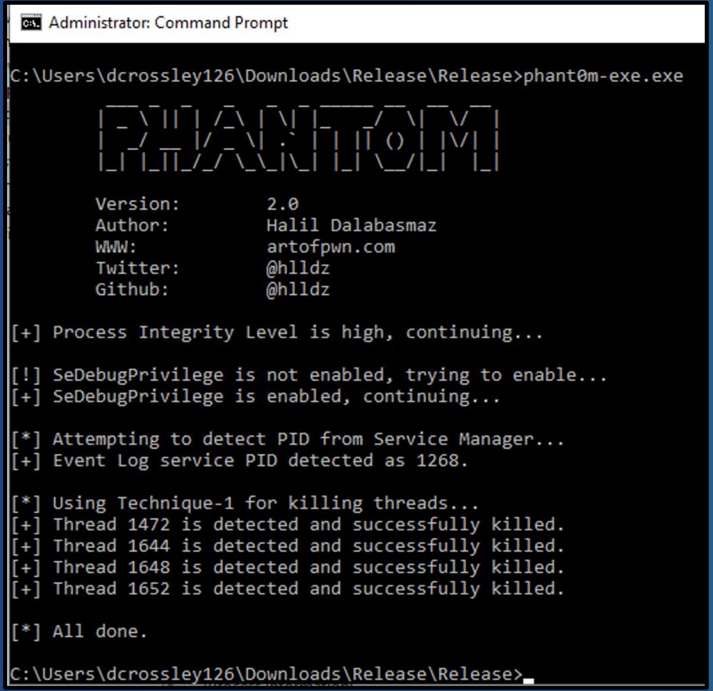
Log Source Disruption

- Missed detections due to log processing issues is a common issue with SIEMs
- Some basic agent interference strategies could be:
    1. Disable SIEM agent service
    2. Perform action
    3. Clear Windows Event Log
    4. Restart agent (within heartbeat check timeframe).
- Or:
    1. Suspend the Windows Event Log
    2. Perform Action
    3. Unsuspend Event Log



Name	Full Name	Description
Administrator		Built-in account for administering...
dan1	dan1	
DefaultAccount		A user account managed by the s...
Guest		Built-in account for guest access t...
vadmin		
WDAGUtilityAccount		A user account managed and use...

Log Source Disruption

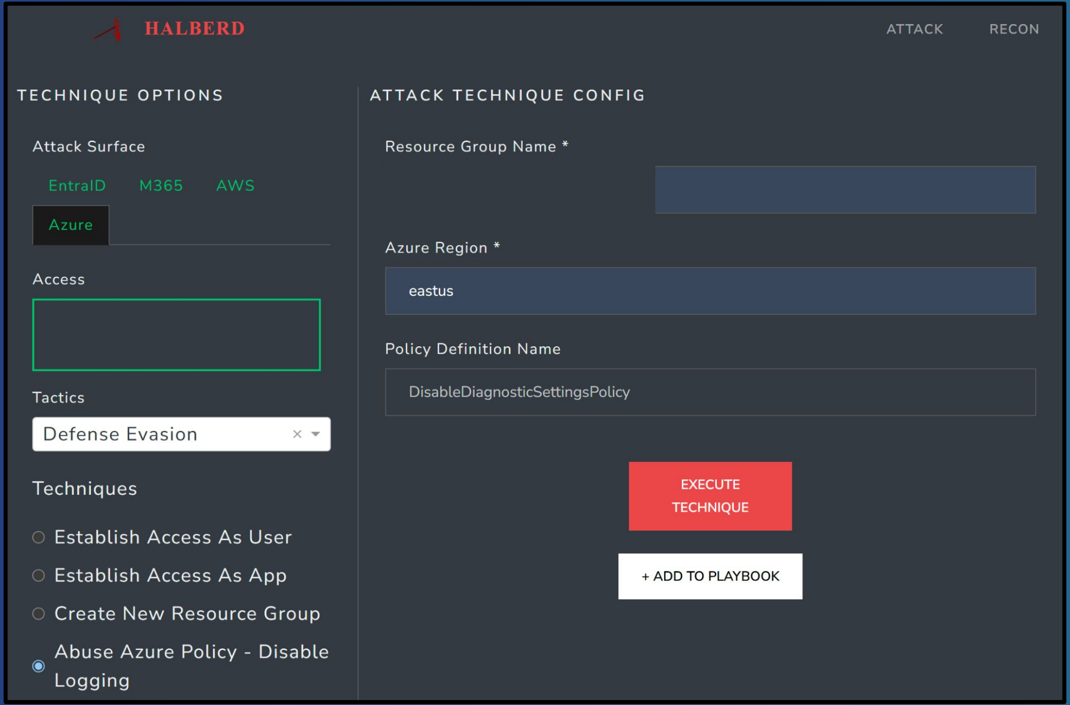


```
C:\Users\dcrossley126\Downloads\Release\Release>phant0m-exe.exe

         ___ _  _   _   _  _ _____ ___  __  __
        | _ \ || | /_\ | \| |_   _/ _ \|  \/  |
        |  _/ __ |/ _ \| .` | | || (_) | |\/| |
        |_| |_||_/_/ \_\_|\_| |_| \___/|_|  |_|

        Version:        2.0
        Author:         Halil Dalabasmaz
        WWW:            artofpwn.com
        Twitter:        @hlldz
        Github:         @hlldz

[+] Process Integrity Level is high, continuing...

[!] SeDebugPrivilege is not enabled, trying to enable...
[+] SeDebugPrivilege is enabled, continuing...

[*] Attempting to detect PID from Service Manager...
[+] Event Log service PID detected as 1268.

[*] Using Technique-1 for killing threads...
[+] Thread 1472 is detected and successfully killed.
[+] Thread 1644 is detected and successfully killed.
[+] Thread 1648 is detected and successfully killed.
[+] Thread 1652 is detected and successfully killed.

[*] All done.

C:\Users\dcrossley126\Downloads\Release\Release>_
```

- Clearing Windows Event Log causes EVID1100. This could trigger a SIEM alert
- The Windows Event Log can also be suspended..
- SIEM can alert on no data, however the standard Sentinel analytic rule only checks for no heartbeat for past hour (for example..)

Source: https://github.com/hlldz/Phant0m

Cloud Logging Disruption



TECHNIQUE OPTIONS

Attack Surface

EntraID M365 AWS Azure

Access

Tactics

Defense Evasion

Techniques

○ Establish Access As User

○ Establish Access As App

○ Create New Resource Group

● Abuse Azure Policy - Disable Logging

ATTACK TECHNIQUE CONFIG

Resource Group Name *

Azure Region *

eastus

Policy Definition Name

DisableDiagnosticSettingsPolicy

Source: https://github.com/vectra-ai-research/Halberd

- Cloud logs can be interrupted or stopped
- Important to understand when this happens
- Use testing tools, such as Halberd (shown) to understand if you have detection coverage for this technique

Log Parsing

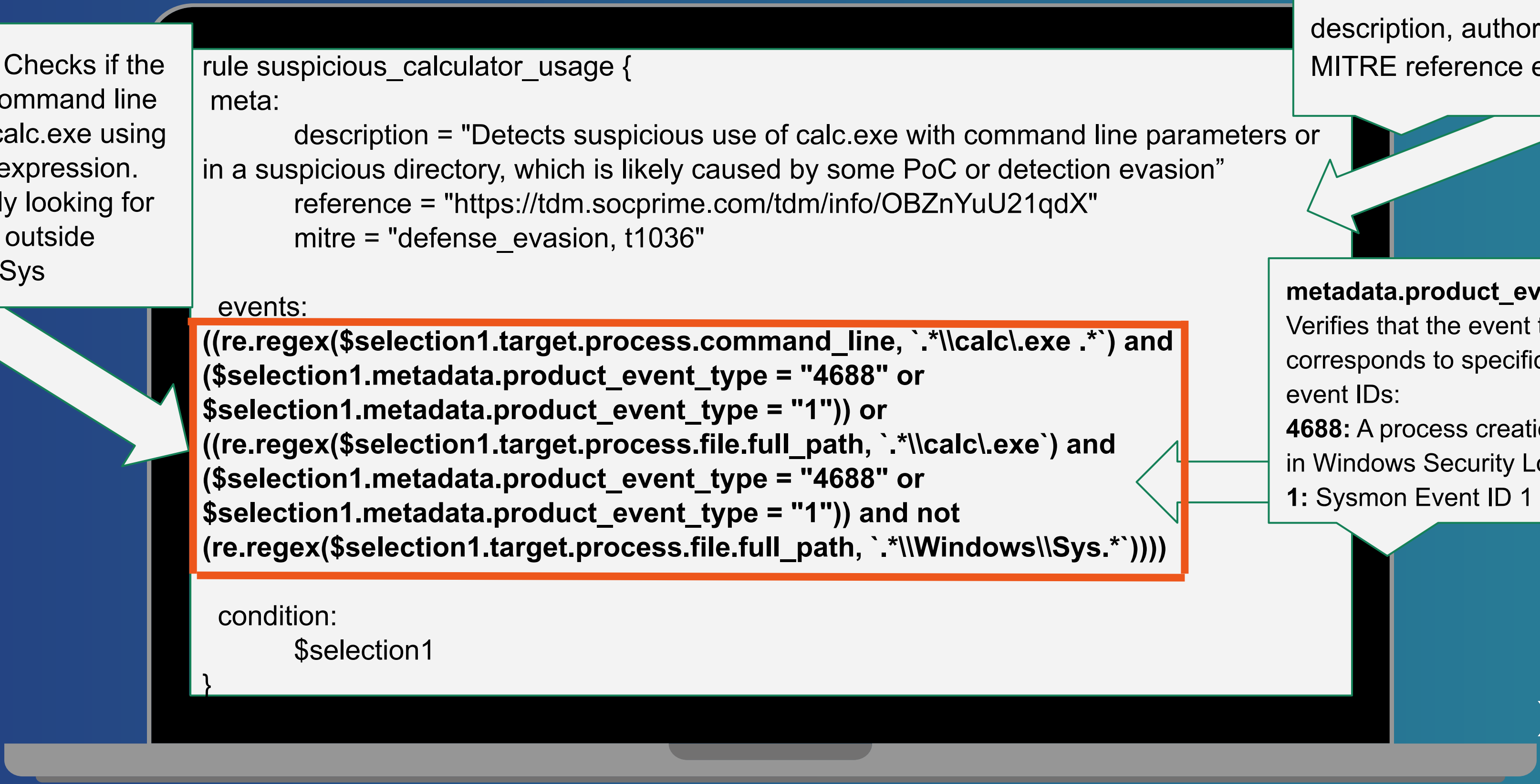
- SIEMs parse logs by applying regular expressions to the raw log data
  - This powers analytic rules, dashboards, reports, etc.
  - Collection time vs Query time
- Parsing can be problematic!
  - Not all fields are correctly parsed
  - Not all logs are categorised correctly
- Creating and maintaining parsing rules can be very complex and time consuming

SIEM RULES

- Where do rules come from?
- In most cases SIEM providers only provide basic rule set leaving the users to create their own.
- Usually turn to community based rules or 3rd party suppliers
- We can leverage this to determine what rules are likely deployed for a given environment

- Ref: https://controlcompass.github.io/

SIEM RULES



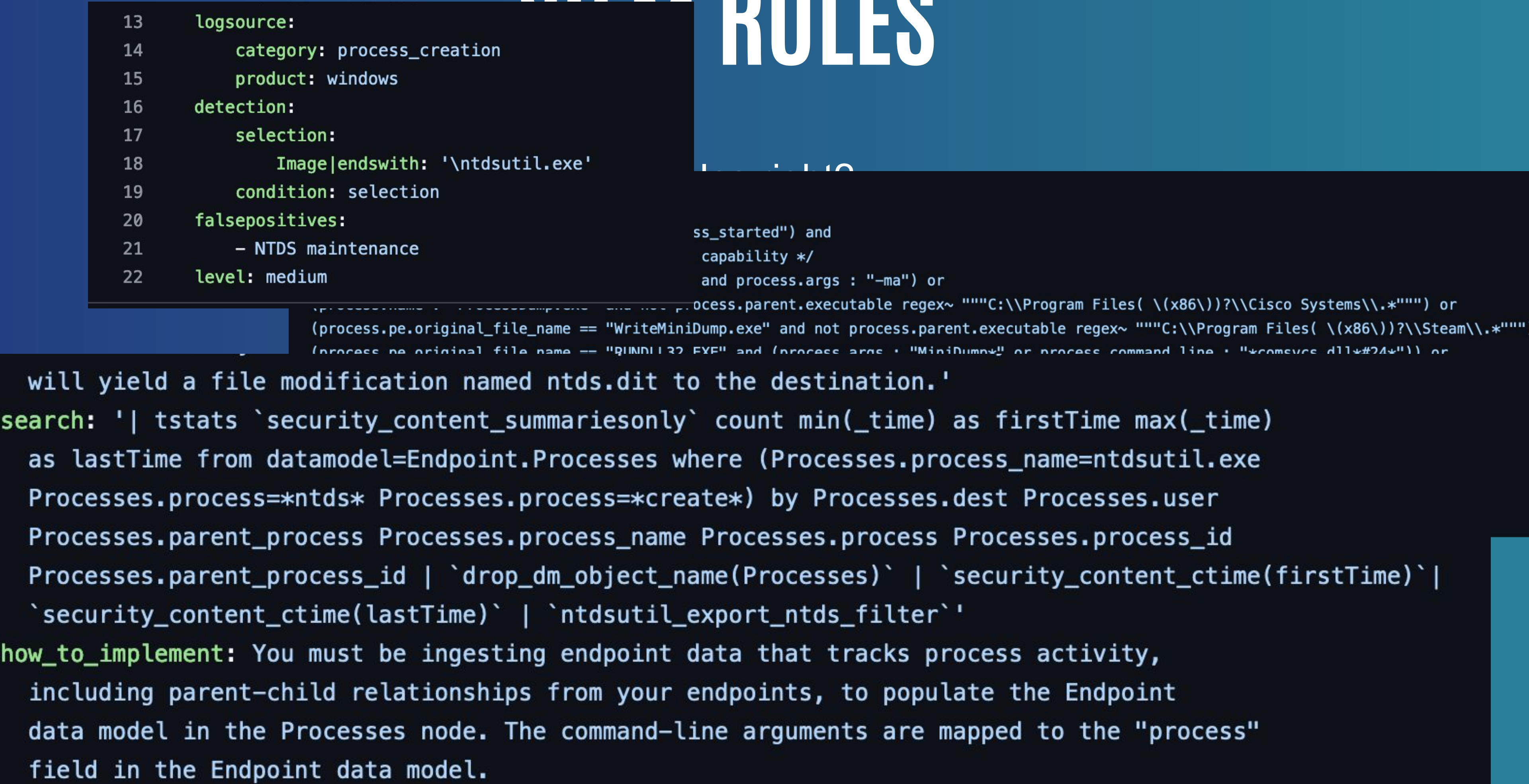
```
rule suspicious_calculator_usage {
 meta:
          description = "Detects suspicious use of calc.exe with command line parameters or
in a suspicious directory, which is likely caused by some PoC or detection evasion”
          reference = "https://tdm.socprime.com/tdm/info/OBZnYuU21qdX"
          mitre = "defense_evasion, t1036"

  events:
((re.regex($selection1.target.process.command_line, `.*\\calc\.exe .*`) and
($selection1.metadata.product_event_type = "4688" or
$selection1.metadata.product_event_type = "1")) or
((re.regex($selection1.target.process.file.full_path, `.*\\calc\.exe`) and
($selection1.metadata.product_event_type = "4688" or
$selection1.metadata.product_event_type = "1")) and not
(re.regex($selection1.target.process.file.full_path, `.*\\Windows\\Sys.*`))))

  condition:
          $selection1
}
```

re.regex: Checks if the process command line contains calc.exe using a regular expression. Specifically looking for execution outside Windows Sys

Metadata for the rule, description, author, MITRE reference etc

metadata.product_event_type: Verifies that the event type corresponds to specific Windows event IDs:
4688: A process creation event in Windows Security Logs.
1: Sysmon Event ID 1

SIEM RULES



```
logsource:
    category: process_creation
    product: windows
detection:
    selection:
        Image|endswith: '\ntdsutil.exe'
    condition: selection
falsepositives:
    - NTDS maintenance
level: medium
```

```
ss_started") and
 capability */
 and process.args : "-ma") or
ocess.parent.executable regex~ """C:\\Program Files( \(x86\))?\\Cisco Systems\\.*""") or
(process.pe.original_file_name == "WriteMiniDump.exe" and not process.parent.executable regex~ """C:\\Program Files( \(x86\))?\\Steam\\.*"""
(process.pe.original_file_name == "RUNDLL32.EXE" and (process.args : "MiniDump*" or process.command_line : "*comsvcs.dll*#24*")) or
```

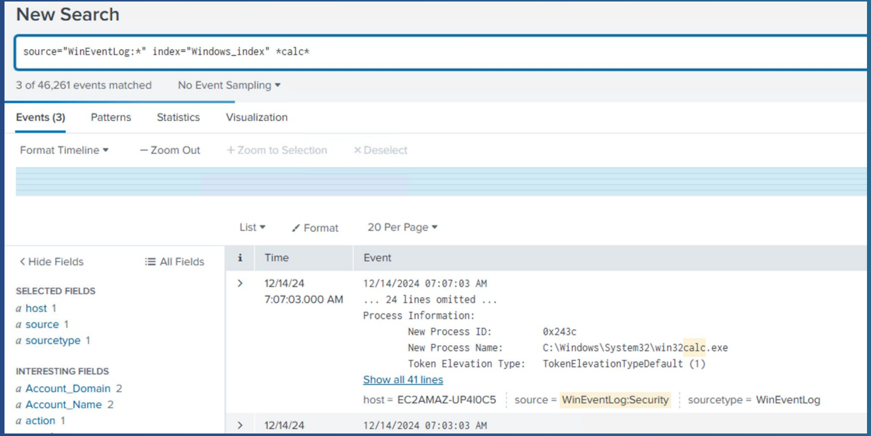
```
  will yield a file modification named ntds.dit to the destination.'
search: '| tstats `security_content_summariesonly` count min(_time) as firstTime max(_time)
  as lastTime from datamodel=Endpoint.Processes where (Processes.process_name=ntdsutil.exe
  Processes.process=*ntds* Processes.process=*create*) by Processes.dest Processes.user
  Processes.parent_process Processes.process_name Processes.process Processes.process_id
  Processes.parent_process_id | `drop_dm_object_name(Processes)` | `security_content_ctime(firstTime)`|
  `security_content_ctime(lastTime)` | `ntdsutil_export_ntds_filter`'
how_to_implement: You must be ingesting endpoint data that tracks process activity,
  including parent-child relationships from your endpoints, to populate the Endpoint
  data model in the Processes node. The command-line arguments are mapped to the "process"
  field in the Endpoint data model.
```

SIEM RULES

Identify the SIEM and review public rules, with this information we can determine alternative methods to invoke an executable and bypass the static rule logic.

Invoke examples

- **Copy executable to a new filename/path**
- **Indirectly invoke the executable, for example with Powershell in memory**
  - **Powershell will still trigger a log, need to use in memory capabilities such as Powerspoint's Invoke-ReflectivePEInjection**
- **Obfuscate the command line**
- **Disable the logging for process creation**



```
source="WinEventLog:*" index="Windows_index" *calc*
```

12/14/24 7:07:03.000 AM

```
12/14/2024 07:07:03 AM
... 24 lines omitted ...
Process Information:
        New Process ID:         0x243c
        New Process Name:       C:\Windows\System32\win32calc.exe
        Token Elevation Type:   TokenElevationTypeDefault (1)
```

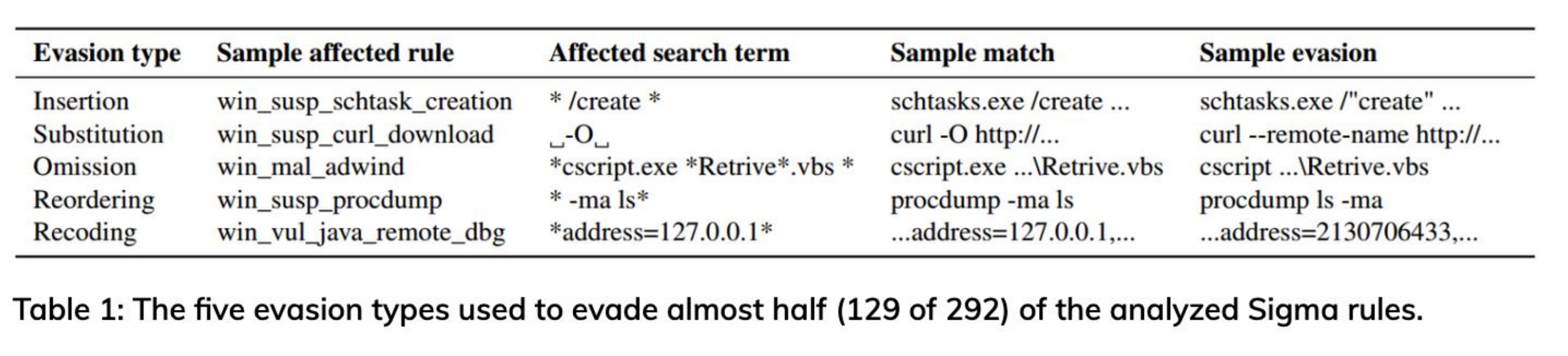
host = EC2AMAZ-UP4I0C5 | source = WinEventLog:Security | sourcetype = WinEventLog

12/14/24

```
12/14/2024 07:03:03 AM
```

Ref: https://powersploit.readthedocs.io/en/latest/CodeExecution/Invoke-ReflectivePEInjection/

SIEM RULES



Evasion type	Sample affected rule	Affected search term	Sample match	Sample evasion
Insertion	win_susp_schtask_creation	* /create *	schtasks.exe /create ...	schtasks.exe /"create" ...
Substitution	win_susp_curl_download	␣-O␣	curl -O http://...	curl --remote-name http://...
Omission	win_mal_adwind	*cscript.exe *Retrive*.vbs *	cscript.exe ...\Retrive.vbs	cscript ...\Retrive.vbs
Reordering	win_susp_procdump	* -ma ls*	procdump -ma ls	procdump ls -ma
Recoding	win_vul_java_remote_dbg	*address=127.0.0.1*	...address=127.0.0.1,...	...address=2130706433,...

Table 1: The five evasion types used to evade almost half (129 of 292) of the analyzed Sigma rules.

Ref: https://www.usenix.org/publications/loginonline/you-cannot-escape-me-detecting-evasions-siem-rules-enterprise-networks

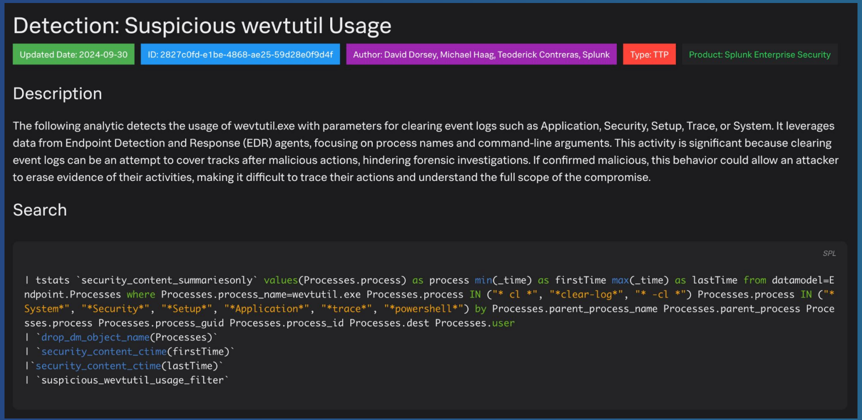
SIEM RULES

But wait my SIEM checks for policy tampering and log clearing…..

- 4907 (audit policy changes)
- 1102 (log clearing)
- 4688 (process creation logging)

This can be challenging to overcome, however there are some techniques which might work (Again you'll need to check the specific SIEM rules & surface you're attacking)

- wevtutil sl Security /q:"Event[System[(EventID=4907)]]"
- auditpol /set /subcategory:"Process Creation" /success:disable /failure:disable
    - (Local admin required for both)



Detection: Suspicious wevtutil Usage

Updated Date: 2024-09-30 | ID: 2827c0fd-e1be-4868-ae25-59d28e0f9d4f | Author: David Dorsey, Michael Haag, Teoderick Contreras, Splunk | Type: TTP | Product: Splunk Enterprise Security

Description

The following analytic detects the usage of wevtutil.exe with parameters for clearing event logs such as Application, Security, Setup, Trace, or System. It leverages data from Endpoint Detection and Response (EDR) agents, focusing on process names and command-line arguments. This activity is significant because clearing event logs can be an attempt to cover tracks after malicious actions, hindering forensic investigations. If confirmed malicious, this behavior could allow an attacker to erase evidence of their activities, making it difficult to trace their actions and understand the full scope of the compromise.

Search

```
| tstats `security_content_summariesonly` values(Processes.process) as process min(_time) as firstTime max(_time) as lastTime from datamodel=Endpoint.Processes where Processes.process_name=wevtutil.exe Processes.process IN ("* cl *", "*clear-log*", "* -cl *") Processes.process IN ("*System*", "*Security*", "*Setup*", "*Application*", "*trace*", "*powershell*") by Processes.parent_process_name Processes.parent_process Processes.process Processes.process_guid Processes.process_id Processes.dest Processes.user
| `drop_dm_object_name(Processes)`
| `security_content_ctime(firstTime)`
|`security_content_ctime(lastTime)`
| `suspicious_wevtutil_usage_filter`
```

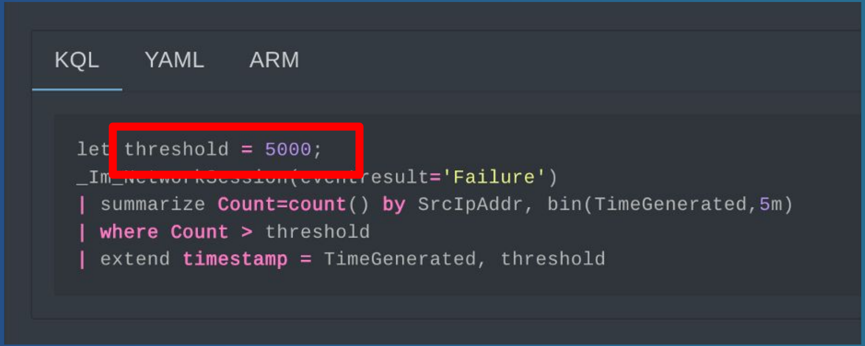
Ref: https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/wevtutil

SIEM RULES

Azure Active Directory	1
CrowdStrike ProcessRollup2, Sysmon EventID 1, Sysmon EventID 12, Sysmon EventID 13, Windows Event Log Security 4688	1
CrowdStrike ProcessRollup2, Sysmon EventID 1, Windows Event Log Security 4688	38
Powershell Installed IIS Modules	1
Powershell Script Block Logging 4104	103
Sysmon EventID 1	2
Sysmon EventID 1, Windows Event Log Security 4688	1
Sysmon EventID 8	1
Windows Event Log Security 4648	1
Windows Event Log System 7045	1

SIEM Rules Bypass Subvert detection logic

- Many SIEM deployments use out of the box correlation / analytic rules
- Many are based on watchlists or asset lists which are often not updated
- Many are simple IOC matches
- Results in minimal detection coverage + bypass techniques!

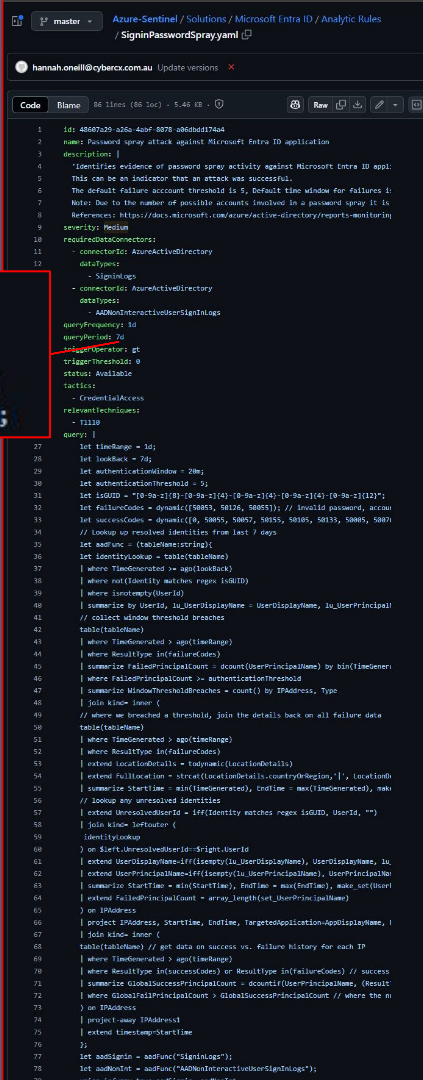


```
let threshold = 5000;
_Im_NetworkSession(eventresult='Failure')
| summarize Count=count() by SrcIpAddr, bin(TimeGenerated,5m)
| where Count > threshold
| extend timestamp = TimeGenerated, threshold
```

SIEM Rules Bypass Password Spray

- Many SIEM rules use static detection logic, i.e. manual thresholds
- E.g. a Password Spraying rule needs balance the observation within a timeframe
- The rule from MS Azure-Sentinel Github repo looks for 5 failed logins for Entra ID from the same IP in 20 mins
- With our new knowledge of SIEM deployment profiles, we know the likelihood of this rule having been tuned..

```
let timeRange = 1d;
let lookBack = 7d;
let authenticationWindow = 20m;
let authenticationThreshold = 5;
```

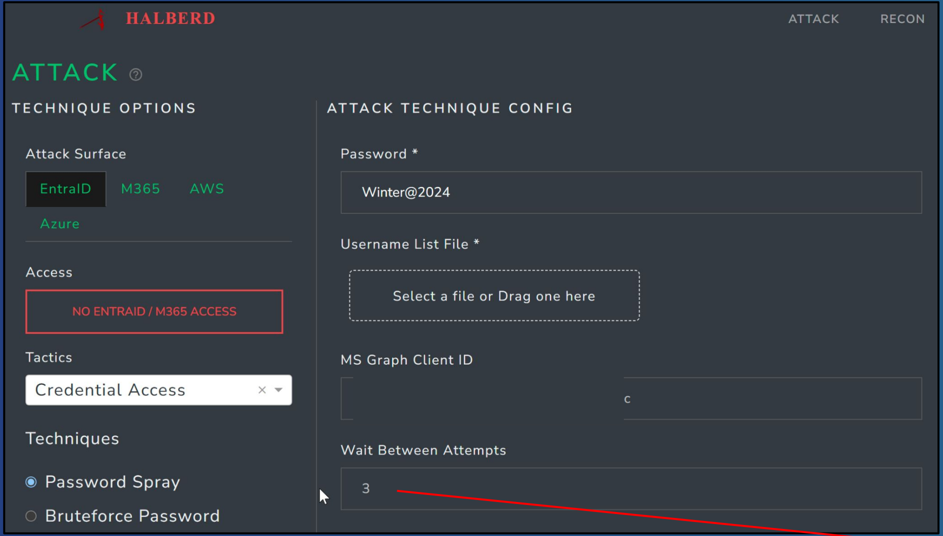


master
Azure-Sentinel / Solutions / Microsoft Entra ID / Analytic Rules / SigninPasswordSpray.yaml
hannah.oneill@cybercx.com.au Update versions
Code
Blame
86 lines (86 loc) · 5.46 KB ·
Raw

```
id: 48607a29-a26a-4abf-8078-a06dbdd174a4
name: Password spray attack against Microsoft Entra ID application
description: |
  'Identifies evidence of password spray activity against Microsoft Entra ID appl:
  This can be an indicator that an attack was successful.
  The default failure acccount threshold is 5, Default time window for failures i:
  Note: Due to the number of possible accounts involved in a password spray it is
  References: https://docs.microsoft.com/azure/active-directory/reports-monitorin;
severity: Medium
requiredDataConnectors:
  - connectorId: AzureActiveDirectory
    dataTypes:
      - SigninLogs
  - connectorId: AzureActiveDirectory
    dataTypes:
      - AADNonInteractiveUserSignInLogs
queryFrequency: 1d
queryPeriod: 7d
triggerOperator: gt
triggerThreshold: 0
status: Available
tactics:
  - CredentialAccess
relevantTechniques:
  - T1110
query: |
    let timeRange = 1d;
    let lookBack = 7d;
    let authenticationWindow = 20m;
    let authenticationThreshold = 5;
    let isGUID = "[0-9a-z]{8}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{12}";
    let failureCodes = dynamic([50053, 50126, 50055]); // invalid password, accou
    let successCodes = dynamic([0, 50055, 50057, 50155, 50105, 50133, 50005, 5007
    // Lookup up resolved identities from last 7 days
    let aadFunc = (tableName:string){
    let identityLookup = table(tableName)
    | where TimeGenerated >= ago(lookBack)
    | where not(Identity matches regex isGUID)
    | where isnotempty(UserId)
    | summarize by UserId, lu_UserDisplayName = UserDisplayName, lu_UserPrincipal
    // collect window threshold breaches
    table(tableName)
    | where TimeGenerated > ago(timeRange)
    | where ResultType in(failureCodes)
    | summarize FailedPrincipalCount = dcount(UserPrincipalName) by bin(TimeGener
    | where FailedPrincipalCount >= authenticationThreshold
    | summarize WindowThresholdBreaches = count() by IPAddress, Type
    | join kind= inner (
    // where we breached a threshold, join the details back on all failure data
    table(tableName)
    | where TimeGenerated > ago(timeRange)
    | where ResultType in(failureCodes)
    | extend LocationDetails = todynamic(LocationDetails)
    | extend FullLocation = strcat(LocationDetails.countryOrRegion,'|', LocationD
    | summarize StartTime = min(TimeGenerated), EndTime = max(TimeGenerated), mak
    // lookup any unresolved identities
    | extend UnresolvedUserId = iff(Identity matches regex isGUID, UserId, "")
    | join kind= leftouter (
     identityLookup
    ) on $left.UnresolvedUserId==$right.UserId
    | extend UserDisplayName=iff(isempty(lu_UserDisplayName), UserDisplayName, lu_
    | extend UserPrincipalName=iff(isempty(lu_UserPrincipalName), UserPrincipalNa
    | summarize StartTime = min(StartTime), EndTime = max(EndTime), make_set(User
    | extend FailedPrincipalCount = array_length(set_UserPrincipalName)
    ) on IPAddress
    | project IPAddress, StartTime, EndTime, TargetedApplication=AppDisplayName,
    | join kind= inner (
    table(tableName) // get data on success vs. failure history for each IP
    | where TimeGenerated > ago(timeRange)
    | where ResultType in(successCodes) or ResultType in(failureCodes) // success
    | summarize GlobalSuccessPrincipalCount = dcountif(UserPrincipalName, (Result
    | where GlobalFailPrincipalCount > GlobalSuccessPrincipalCount // where the n
    ) on IPAddress
    | project-away IPAddress1
    | extend timestamp=StartTime
    };
    let aadSignin = aadFunc("SigninLogs");
    let aadNonInt = aadFunc("AADNonInteractiveUserSignInLogs");
```

Source:
https://github.com/Azure/Azure-Sentinel/blob/master/Solutions/Microsoft%20Entra%20ID/Analytic%20Rules/SigninPasswordSpray.yaml

SIEM Rules Bypass Password Spray



ATTACK

TECHNIQUE OPTIONS

Attack Surface

EntraID M365 AWS Azure

Access

NO ENTRAID / M365 ACCESS

Tactics

Credential Access

Techniques

● Password Spray

○ Bruteforce Password

ATTACK TECHNIQUE CONFIG

Password *

Winter@2024

Username List File *

Select a file or Drag one here

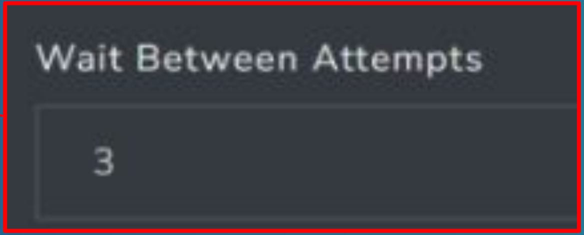
MS Graph Client ID

[redacted]c

Wait Between Attempts

3

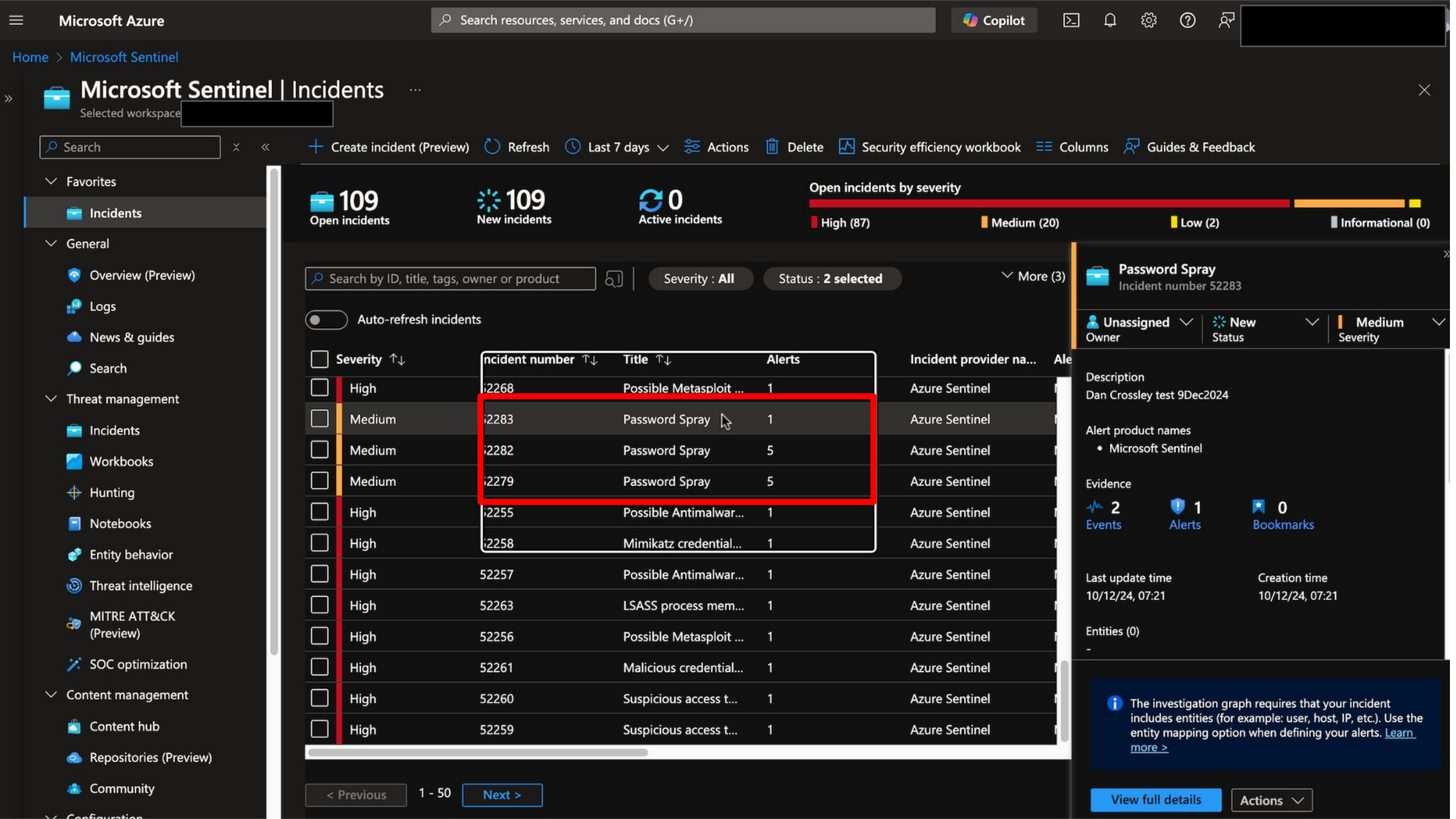
- We can test EntraID password spray using tool ‘Halberd’
- Using default spray of 3 seconds between attempts, the Sentinel Analytic rule triggers as expected



Wait Between Attempts

3

Source: https://github.com/vectra-ai-research/Halberd


Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
[redacted]
Home > Microsoft Sentinel

Microsoft Sentinel | Incidents

Selected workspace [redacted]

109 Open incidents

109 New incidents

0 Active incidents

Open incidents by severity

High (87) Medium (20) Low (2) Informational (0)

	Severity	ncident number	Title	Alerts	Incident provider na...	Ale
□	High	2268	Possible Metasploit ...	1	Azure Sentinel	
□	Medium	2283	Password Spray	1	Azure Sentinel	
□	Medium	2282	Password Spray	5	Azure Sentinel	
□	Medium	2279	Password Spray	5	Azure Sentinel	
□	High	2255	Possible Antimalwar...	1	Azure Sentinel	
□	High	2258	Mimikatz credential...	1	Azure Sentinel	
□	High	52257	Possible Antimalwar...	1	Azure Sentinel	
□	High	52263	LSASS process mem...	1	Azure Sentinel	
□	High	52256	Possible Metasploit ...	1	Azure Sentinel	
□	High	52261	Malicious credential...	1	Azure Sentinel	
□	High	52260	Suspicious access t...	1	Azure Sentinel	
□	High	52259	Suspicious access t...	1	Azure Sentinel	

Password Spray
Incident number 52283

Owner: Unassigned
Status: New
Severity: Medium

Description
Dan Crossley test 9Dec2024

Alert product names
- Microsoft Sentinel

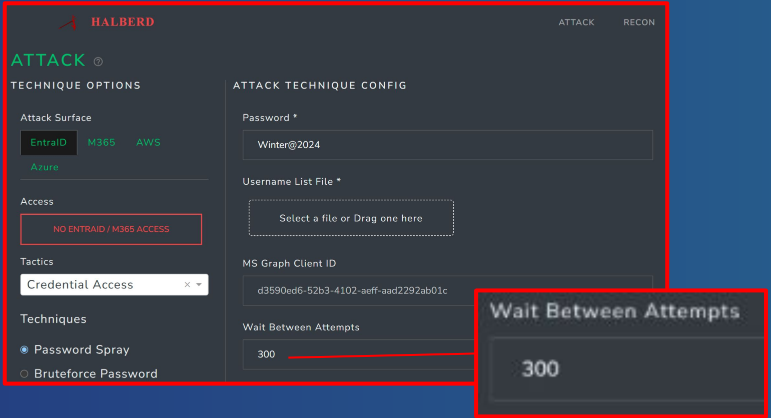
Evidence
2 Events
1 Alerts
0 Bookmarks

Last update time: 10/12/24, 07:21
Creation time: 10/12/24, 07:21

Entities (0)
-

The investigation graph requires that your incident includes entities (for example: user, host, IP, etc.). Use the entity mapping option when defining your alerts. Learn more >

SIEM Rules Bypass Password Spray



ATTACK

TECHNIQUE OPTIONS

Attack Surface: EntraID | M365 | AWS | Azure

Access: NO ENTRAID / M365 ACCESS

Tactics: Credential Access

Techniques

- ● Password Spray
- ○ Bruteforce Password

ATTACK TECHNIQUE CONFIG

Password *: Winter@2024

Username List File *: Select a file or Drag one here

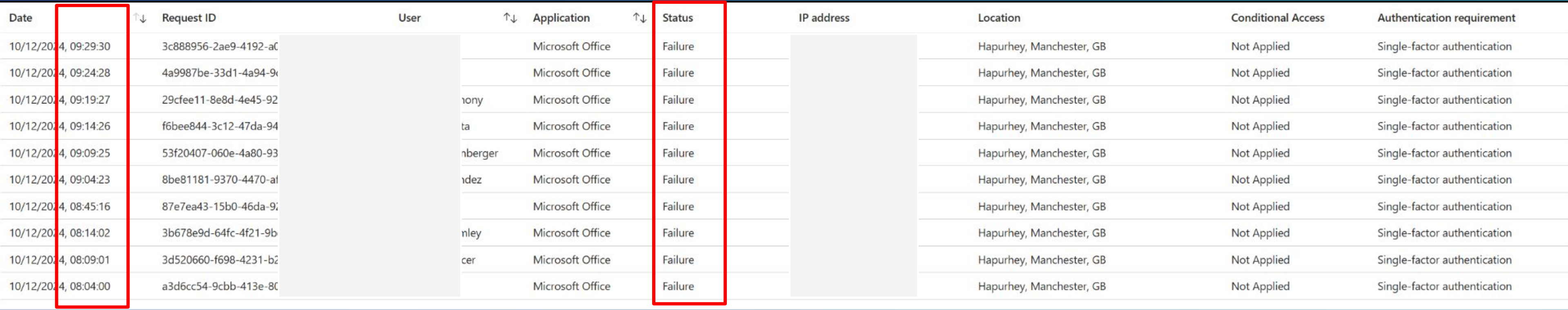
MS Graph Client ID: d3590ed6-52b3-4102-aeff-aad2292ab01c

Wait Between Attempts: 300

Wait Between Attempts

300

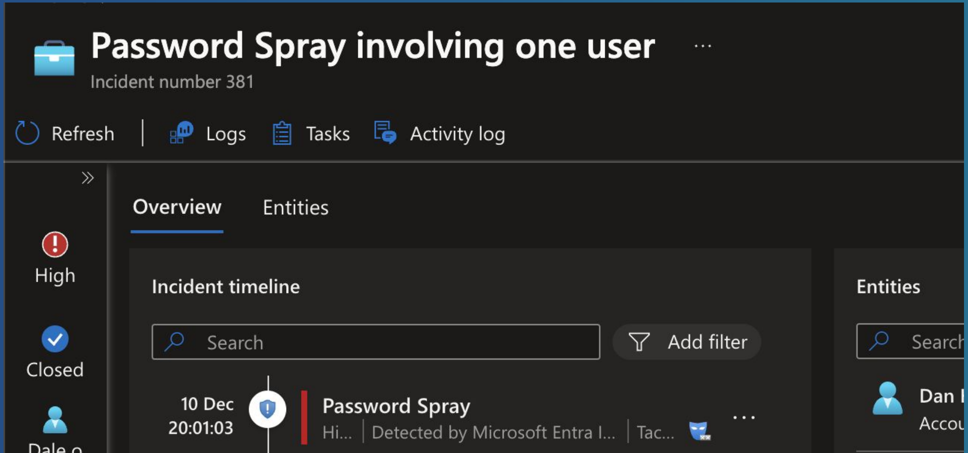
- Running the same password spraying attack with a wait of 300 seconds (ie 4 attempts every 20 minutes)
- Sentinel Analytic rule does not trigger..



Date	Request ID	User	Application	Status	IP address	Location	Conditional Access	Authentication requirement
10/12/2024, 09:29:30	3c888956-2ae9-4192-a	[redacted]	Microsoft Office	Failure	[redacted]	Hapurhey, Manchester, GB	Not Applied	Single-factor authentication
10/12/2024, 09:24:28	4a9987be-33d1-4a94-9	[redacted]	Microsoft Office	Failure	[redacted]	Hapurhey, Manchester, GB	Not Applied	Single-factor authentication
10/12/2024, 09:19:27	29cfee11-8e8d-4e45-92	[redacted]	Microsoft Office	Failure	[redacted]	Hapurhey, Manchester, GB	Not Applied	Single-factor authentication
10/12/2024, 09:14:26	f6bee844-3c12-47da-94	[redacted]	Microsoft Office	Failure	[redacted]	Hapurhey, Manchester, GB	Not Applied	Single-factor authentication
10/12/2024, 09:09:25	53f20407-060e-4a80-93	[redacted]	Microsoft Office	Failure	[redacted]	Hapurhey, Manchester, GB	Not Applied	Single-factor authentication
10/12/2024, 09:04:23	8be81181-9370-4470-a	[redacted]	Microsoft Office	Failure	[redacted]	Hapurhey, Manchester, GB	Not Applied	Single-factor authentication
10/12/2024, 08:45:16	87e7ea43-15b0-46da-9	[redacted]	Microsoft Office	Failure	[redacted]	Hapurhey, Manchester, GB	Not Applied	Single-factor authentication
10/12/2024, 08:14:02	3b678e9d-64fc-4f21-9b	[redacted]	Microsoft Office	Failure	[redacted]	Hapurhey, Manchester, GB	Not Applied	Single-factor authentication
10/12/2024, 08:09:01	3d520660-f698-4231-b2	[redacted]	Microsoft Office	Failure	[redacted]	Hapurhey, Manchester, GB	Not Applied	Single-factor authentication
10/12/2024, 08:04:00	a3d6cc54-9cbb-413e-8	[redacted]	Microsoft Office	Failure	[redacted]	Hapurhey, Manchester, GB	Not Applied	Single-factor authentication

SIEM Rules Bypass Subvert Detection Logic

- Understand not only your detection coverage but the logic behind the detections
  - Sometimes a detection may not cover you as you may think
- A multi-layered detection approach is key



Password Spray involving one user

Incident number 381

10 Dec 20:01:03

Password Spray

Hi... | Detected by Microsoft Entra I... | Tac...

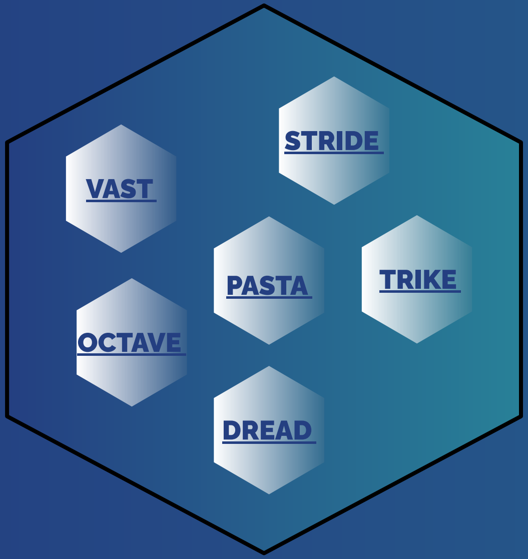
Dan I

Accou

Advice and Mitigations

Mitigations

Threat Modelling



STRIDE

VAST

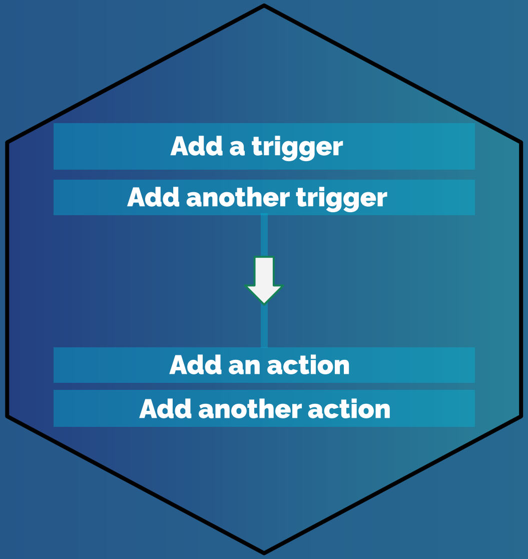
PASTA

TRIKE

OCTAVE

DREAD

Use Case Definition



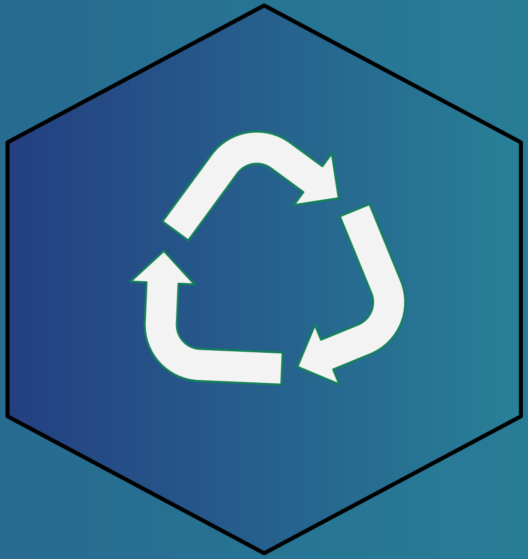
Add a trigger

Add another trigger

Add an action

Add another action

Continuous Improvement



FIN

Questions

.*?