RSAConference2020

San Francisco | February 24 – 28 | Moscone Center

HUMAN ELEMENT

SESSION ID: **AIR-R02**

MITRE ATT&CK - THE SEQUEL

Freddy Dezeure

CEO
Freddy Dezeure BV
@Fdezeure
www.freddydezeure.eu

Rich Struse

Director, Center for Threat-Informed Defense
MITRE Engenuity
@MITREattack
attack.mitre.org

The Sequel

- Presentation builds on our RSA2019 MITRE ATT&CK presentation
- Our goal is to provide real hands-on guidance
- Everything was built in cooperation with Munich Airport

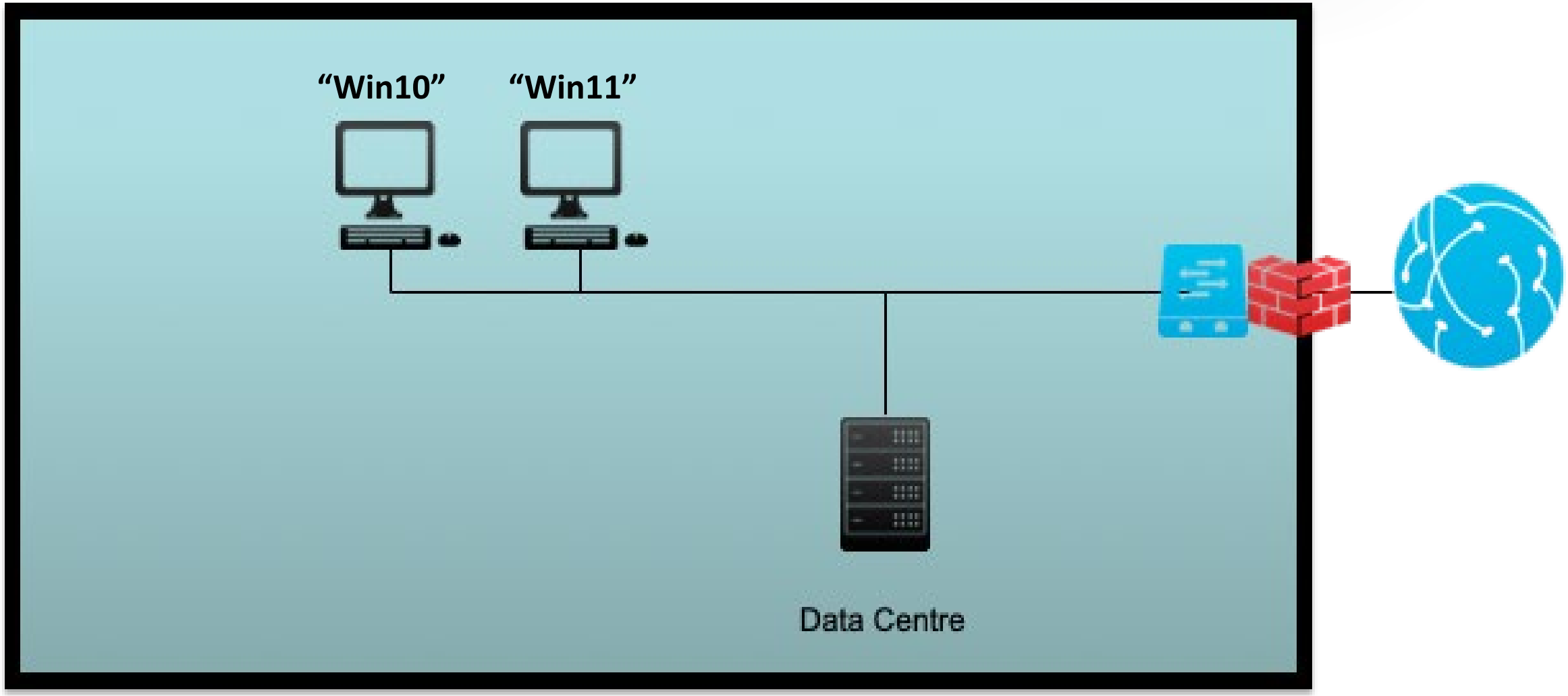
Agenda

- Identify
- Protect
- Detect
- Update
- Share

Our Enterprise Is A Financial Service

- We process money for our clients
- Our main risks:
  - Financial loss
  - Business continuity
  - Brand damage
  - GDPR
- Our infrastructure is well protected (we think)
- We want to perform threat-informed defense

Our Infrastructure



Our Infrastructure

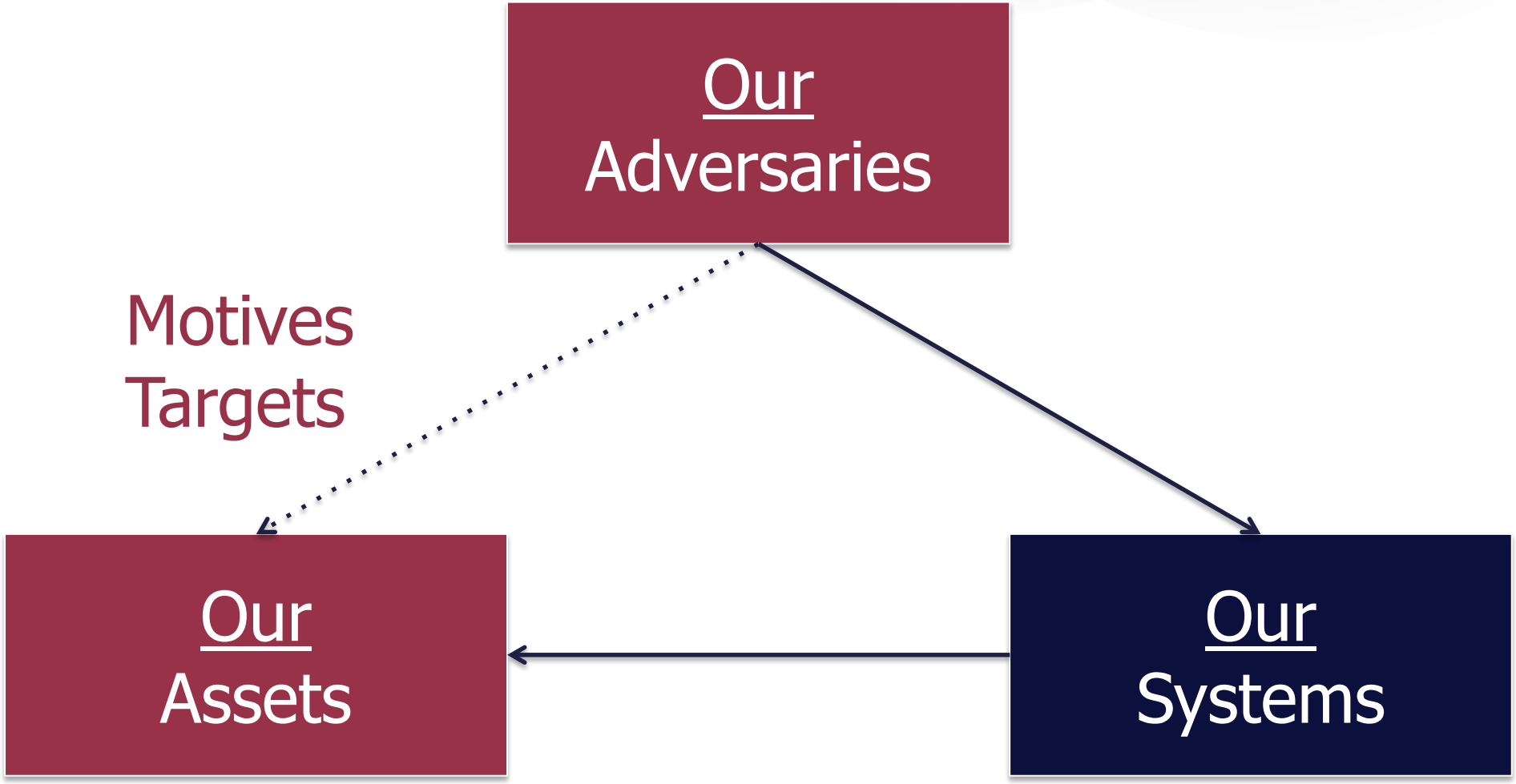
- Created in Detection Lab
  - Installed from GitHub
  - + One additional host
  - + Squid proxy
  - + Caldera
- We populated the logfiles by normal user behavior
- We executed our scenario and made screenshots

Identify

Our Assets, Our Infrastructure, Our Main Adversaries And Their TTPs

Identify Our Adversaries' Objectives And Behavior

- Identify our Adversaries of interest
  - Open source and commercial threat intelligence
  - ISACs/ISAOs
  - NCICC/CERTs
- Identify which tactics/techniques they use
  - ATT&CK Navigator


Our Adversaries
Motives
Targets
Our Assets
Our Systems

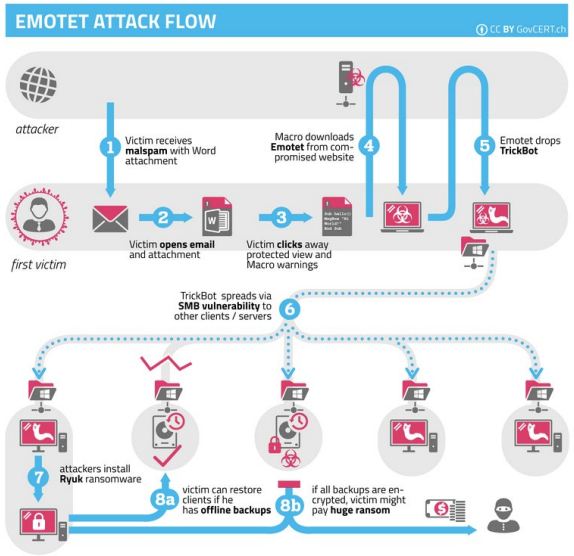
Our Main Adversaries

- Cross-sector : targeted ransomware

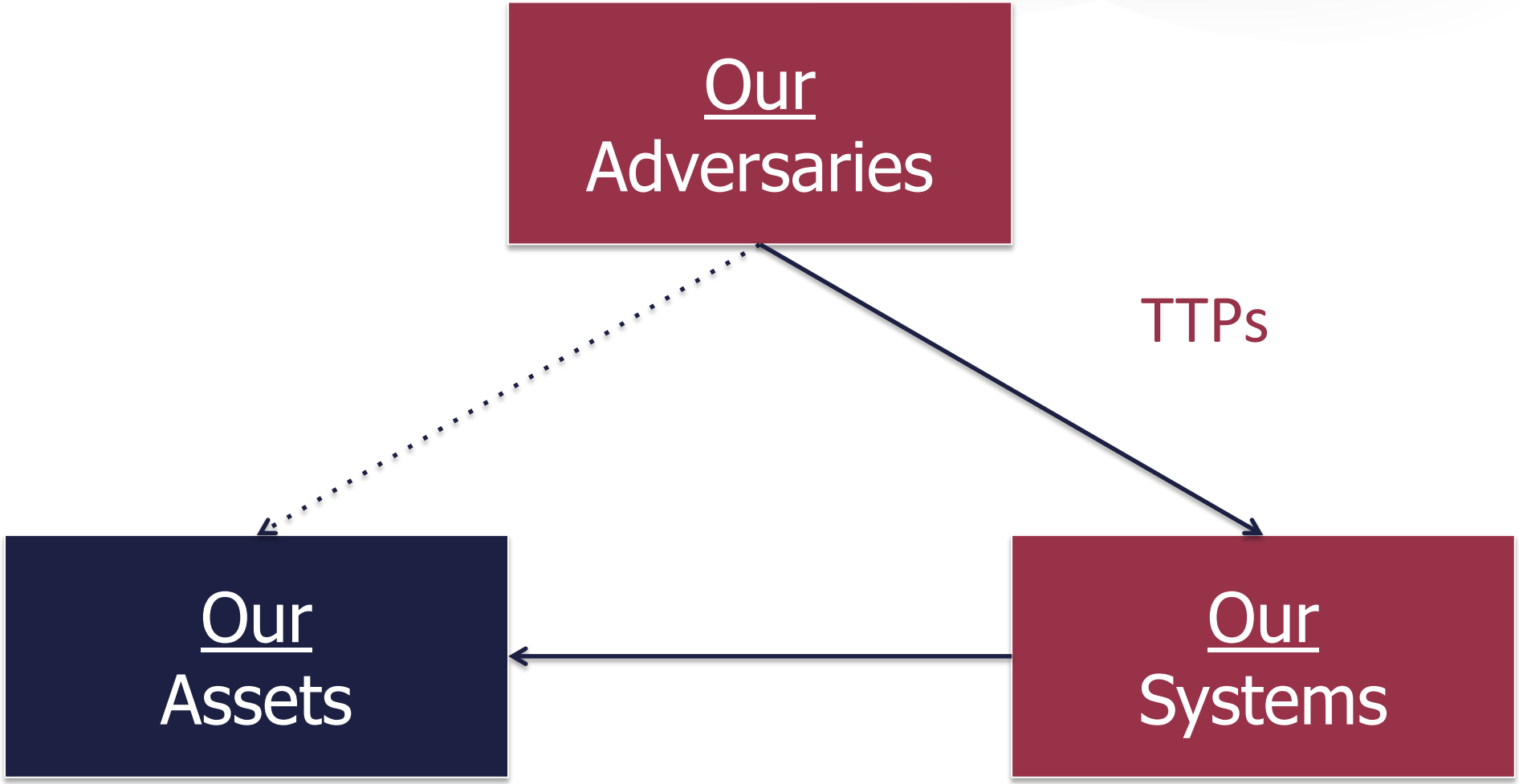
  Emotet

  followed by Trickbot

  Followed by Ryuk/LockerGoga…



- Sectoral : Fin7, Cobalt Group


Our Adversaries
TTPs
Our Assets
Our Systems

Emotet

Emotet is a modular malware variant which is primarily used as a downloader for other malware variants such as TrickBot and IcedID. Emotet first emerged in June 2014 and has been primarily used to target the banking sector. [1]

ID: S0367
Associated Software: Geodo
Type: MALWARE
Platforms: Windows
Contributors: Omkar Gudhate
Version: 1.1
Created: 25 March 2019
Last Modified: 28 June 2019

Associated Software Descriptions

Name	Description
Geodo	[7]

Techniques Used

ATT&CK™ Navigator Layers

Domain	ID	Name	Use
Enterprise	T1110	Brute Force	Emotet has been observed using a hard coded list of passwords to brute force user accounts. [2][3][4][5][6]
Enterprise	T1059	Command-Line Interface	Emotet has used cmd.exe to run a PowerShell script.[9]

Emotet (S0367) x +

selection controls | layer controls | technique controls

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command And Control	Exfiltration	Impact
11 items	34 items	62 items	32 items	69 items	21 items	23 items	18 items	13 items	22 items	9 items	16 items
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
Hardware Additions	Compiled HTML File	AppCert DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	Domain Trust Discovery	Exploitation of Remote Services	Data from Information Repositories	Custom Command and Control Protocol	Data Transfer Size Limits	Defacement
Replication Through Removable Media	Component Object Model and Distributed COM	AppInit DLLs	Application Shimming	Clear Command History	Credentials from Web Browsers	File and Directory Discovery	Internal Spearphishing	Data from Local System	Custom Cryptographic Protocol	Exfiltration Over Alternative Protocol	Disk Content Wipe
Spearphishing Attachment	Control Panel Items	Application Shimming	Bypass User Account Control	CMSTP	Credentials in Files	Network Service Scanning	Logon Scripts	Data from Network Shared Drive	Data Encoding	Exfiltration Over Command and Control Channel	Disk Structure Wipe
Spearphishing Link	Dynamic Data Exchange	Authentication Package	DLL Search Order Hijacking	Code Signing	Credentials in Registry	Network Share Discovery	Pass the Hash	Data from Removable Media	Data Obfuscation	Exfiltration Over Other Network Medium	Endpoint Denial of Service
Spearphishing via Service	Execution through API	BITS Jobs	Dylib Hijacking	Compile After Delivery	Exploitation for Credential Access	Network Sniffing	Pass the Ticket	Data Staged	Domain Fronting	Exfiltration Over Physical Medium	Firmware Corruption
Supply Chain Compromise	Execution through Module Load	Bootkit	Elevated Execution with Prompt	Compiled HTML File	Forced Authentication	Password Policy Discovery	Remote Desktop Protocol	Email Collection	Domain Generation Algorithms	Scheduled Transfer	Inhibit System Recovery
Trusted Relationship	Exploitation for Client Execution	Browser Extensions	Emond	Component Firmware	Hooking	Peripheral Device Discovery	Remote File Copy	Input Capture	Fallback Channels		Network Denial of Service
Valid Accounts	Graphical User Interface	Change Default File Association	Exploitation for Privilege Escalation	Component Object Model Hijacking	Input Capture	Permission Groups Discovery	Remote Services	Man in the Browser	Multi-hop Proxy		Resource Hijacking
	InstallUtil	Component Firmware	Extra Window Memory Injection	Connection Proxy	Input Prompt	Process Discovery	Replication Through Removable Media	Screen Capture	Multi-Stage Channels		Runtime Data Manipulation
	Launchctl	Component Object Model Hijacking	File System Permissions Weakness	Control Panel Items	Kerberoasting	Query Registry	Shared Webroot	Video Capture	Multiband Communication		Service Stop
	Local Job Scheduling	Create Account	Hooking	DCShadow	Keychain	Remote System Discovery	SSH Hijacking		Multilayer Encryption		Stored Data Manipulation
	LSASS Driver	DLL Search Order Hijacking	Image File Execution	Deobfuscate/Decode Files or Information	LLMNR/NBT-NS Poisoning and Relay	Security Software Discovery	Taint Shared Content		Port Knocking		System Shutdown/Reboot
	Mshta	Dylib Hijacking		Disabling Security Tools	Network Sniffing	Software Discovery			Remote Access Tools		Transmitted Data Manipulation
	PowerShell	Emond		DLL Search Order Hijacking	Password Filter DLL	System Information Discovery					
				DLL Side-Loading							
				Execution Guardrails							
				Exploitation for Defense Evasion							

legend

GROUPS
Overview
admin@338
APT1
APT12
APT16
APT17
APT18
APT19
APT28
APT29
APT3
APT30
APT32
APT33
APT37
APT38

Home > Groups > FIN7

FIN7

FIN7 is a financially-motivated threat group that has primarily targeted the U.S. retail, restaurant, and hospitality sectors since mid-2015. They often use point-of-sale malware. A portion of FIN7 was run out of a front company called Combi Security. FIN7 is sometimes referred to as Carbanak Group, but these appear to be two groups using the same Carbanak malware and are therefore tracked separately. [1] [2] [3] [4]

ID: G0046

Version: 1.3

Techniques Used

ATT&CK™ Navigator Layers

Domain	ID	Name	Use
Enterprise	T1138	Application Shimming	FIN7 has used application shim databases for persistence.[7]
Enterprise	T1116	Code Signing	FIN7 has signed Carbanak payloads with legally purchased code signing certificates. FIN7 has also digitally signed their phishing documents, backdoors and other staging tools to bypass security controls.[3][4]
Enterprise	T1059	Command-Line Interface	FIN7 used cmd.exe to launch commands on the victim's machine.[4]
Enterprise	T1043	Commonly Used Port	FIN7 has used ports 53, 80, 443, and 8080 for C2.[4]

GROUPS
Overview
admin@338
APT1
APT12
APT16
APT17
APT18
APT19
APT28
APT29
APT3
APT30
APT32
APT33
APT37
APT38

Home > Groups > Cobalt Group

Cobalt Group

Cobalt Group is a financially motivated threat group that has primarily targeted financial institutions. The group has conducted intrusions to steal money via targeting ATM systems, card processing, payment systems and SWIFT systems. Cobalt Group has mainly targeted banks in Eastern Europe, Central Asia, and Southeast Asia. One of the alleged leaders was arrested in Spain in early 2018, but the group still appears to be active. The group has been known to target organizations in order to use their access to then compromise additional victims. [1] [2] [3] [4] [5] [6] [7] Reporting indicates there may be links between Cobalt Group and both the malware Carbanak and the group Carbanak. [8]

ID: G0080

Associated Groups: Cobalt Gang, Cobalt Spider

Version: 1.1

Associated Group Descriptions

Name	Description
Cobalt Gang	[1] [12][9]
Cobalt Spider	[12]

Techniques Used

ATT&CK™ Navigator Layers

RSAConference2020

Emotet (S0367) x | FIN7 (G0046) x | Cobalt Group (G0080) x | Combined x

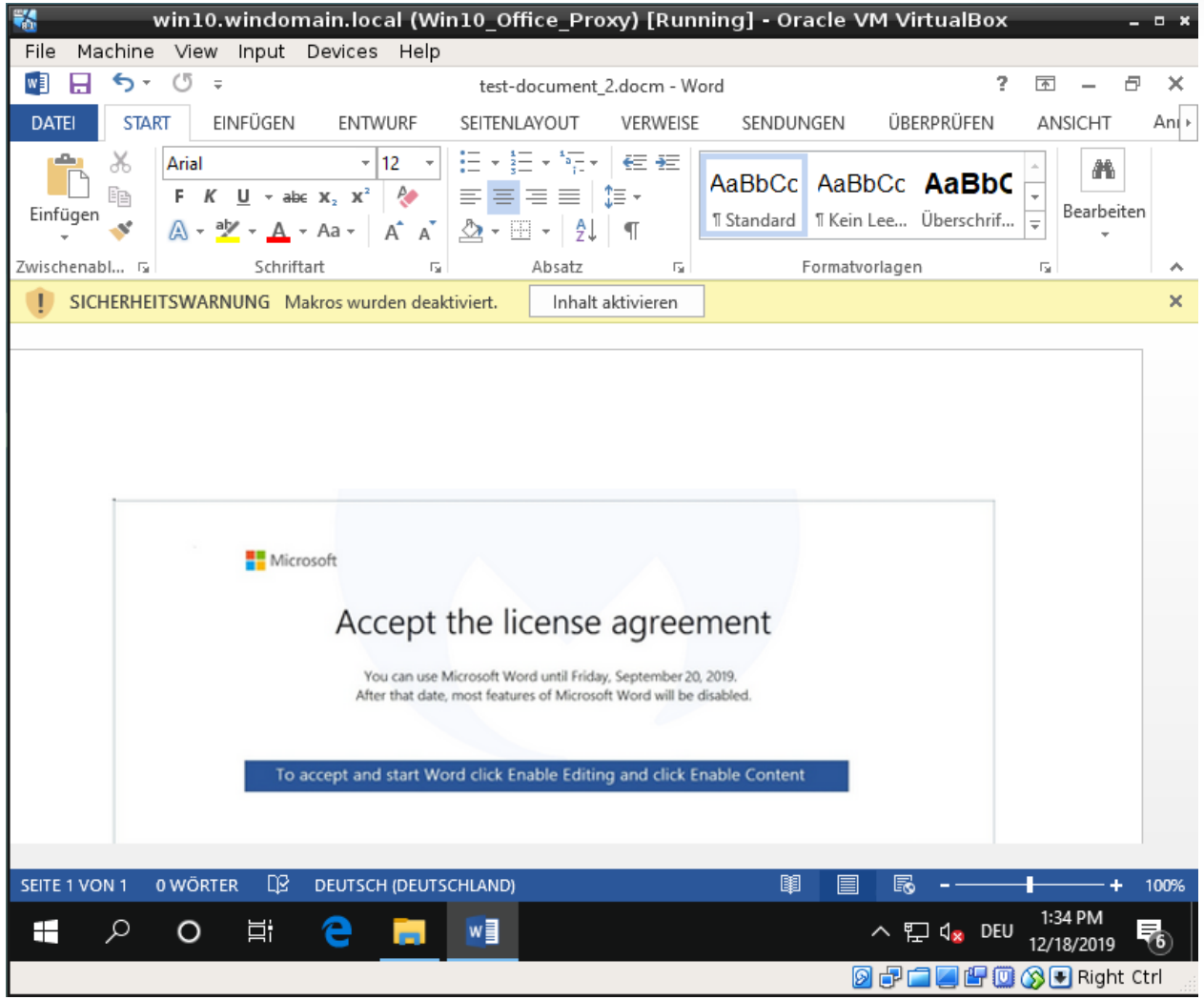
selection controls | layer controls | technique controls

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command And Control	Exfiltration	Impact
11 items	34 items	62 items	32 items	69 items	21 items	23 items	18 items	13 items	22 items	9 items	16 items
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
Hardware Additions	Compiled HTML File	AppCert DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	Domain Trust Discovery	Exploitation of Remote Services	Data from Information Repositories	Custom Command and Control Protocol	Data Transfer Size Limits	Defacement
Replication Through Removable Media	Component Object Model and Distributed COM	AppInit DLLs	Application Shimming	Clear Command History	Credentials from Web Browsers	File and Directory Discovery	Internal Spearphishing	Data from Local System	Custom Cryptographic Protocol	Exfiltration Over Alternative Protocol	Disk Content Wipe
Spearphishing Attachment	Control Panel Items	Application Shimming	Bypass User Account Control	CMSTP	Credentials in Files	Network Service Scanning	Logon Scripts	Data from Network Shared Drive	Data Encoding	Exfiltration Over Command and Control Channel	Disk Structure Wipe
Spearphishing Link	Dynamic Data Exchange	Authentication Package	DLL Search Order Hijacking	Code Signing	Credentials in Registry	Network Share Discovery	Pass the Hash	Data from Removable Media	Data Obfuscation	Exfiltration Over Other Network Medium	Endpoint Denial of Service
Spearphishing via Service	Execution through API	BITS Jobs	Dylib Hijacking	Compile After Delivery	Exploitation for Credential Access	Network Sniffing	Pass the Ticket	Data Staged	Domain Fronting	Exfiltration Over Physical Medium	Firmware Corruption
Supply Chain Compromise	Execution through Module Load	Bootkit	Elevated Execution with Prompt	Compiled HTML File	Forced Authentication	Password Policy Discovery	Remote Desktop Protocol	Email Collection	Domain Generation Algorithms	Scheduled Transfer	Inhibit System Recovery
Trusted Relationship	Exploitation for Client Execution	Browser Extensions	Emond	Component Firmware	Hooking	Peripheral Device Discovery	Remote File Copy	Input Capture	Fallback Channels		Network Denial of Service
Valid Accounts	Graphical User Interface	Change Default File Association	Exploitation for Privilege Escalation	Component Object Model Hijacking	Input Capture	Permission Groups Discovery	Remote Services	Man in the Browser	Multi-hop Proxy		Resource Hijacking
	InstallUtil	Component Firmware	Extra Window Memory Injection	Connection Proxy	Input Prompt	Process Discovery	Replication Through Removable Media	Screen Capture	Multi-Stage Channels		Runtime Data Manipulation
	Launchctl	Component Object Model Hijacking	File System Permissions Weakness	Control Panel Items	Kerberoasting	Query Registry	Shared Webroot	Video Capture	Multiband Communication		Service Stop
	Local Job Scheduling	Create Account	Hooking	DCShadow	Keychain	Remote System Discovery	SSH Hijacking		Multilayer Encryption		Stored Data Manipulation
	LSASS Driver	DLL Search Order Hijacking	Image File Execution	Deobfuscate/Decode Files or Information	LLMNR/NBT-NS Poisoning and Relay	Security Software Discovery	Taint Shared Content		Port Knocking		System Shutdown/Reboot
	Mshta	Dylib Hijacking		Disabling Security Tools	Network Sniffing	Software Discovery			Remote Access Tools		Transmitted Data Manipulation
	PowerShell	Emond		DLL Search Order Hijacking	Password Filter DLL	System Information Discovery					
				DLL Side-Loading							
				Execution Guardrails							
				Exploitation for Defense Evasion							

legend

RSAConference2020

We Built And Used A Realistic Exploit

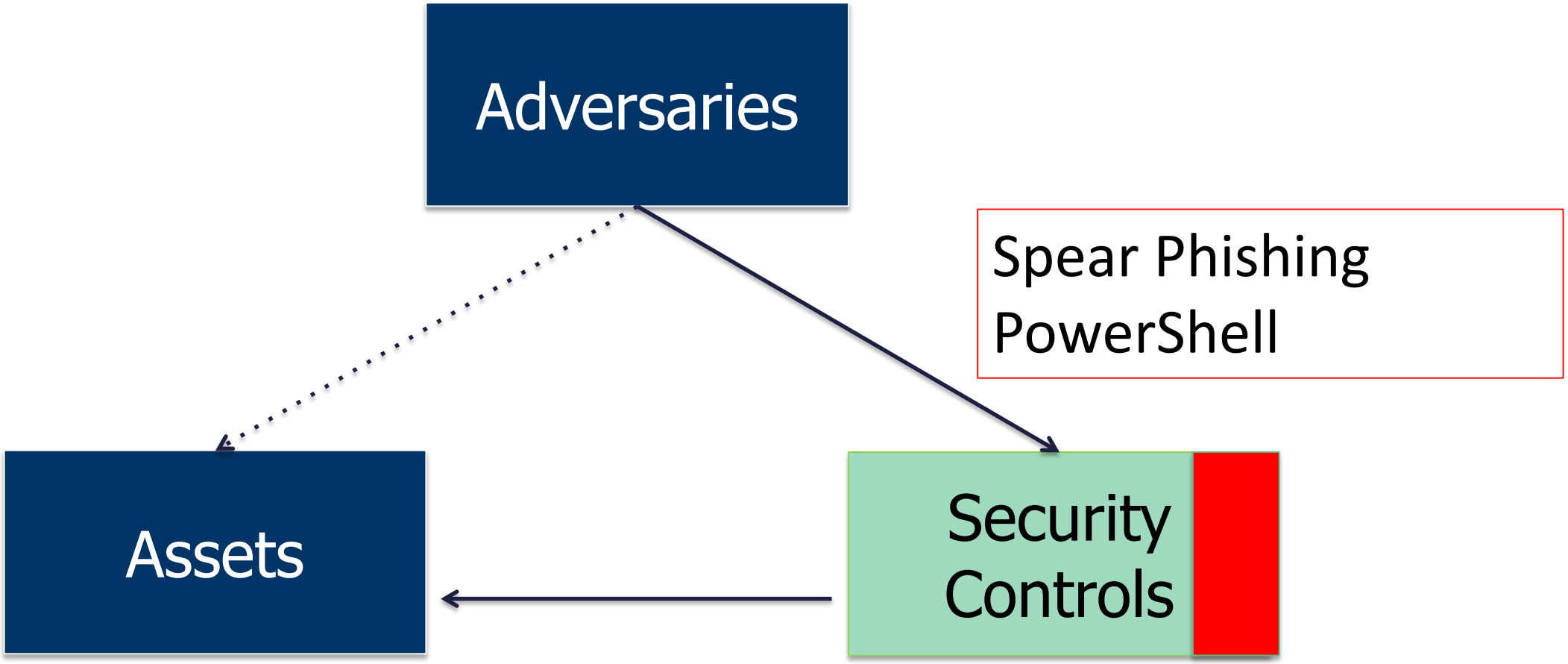


- Word lure document with PowerShell macro connecting to api.ipify.org to grab external ip of our infrastructure and vizualize it

Protect

Design And Validate Our Critical Controls

Design Our Controls



Mitigations For T1086 PowerShell

MITRE ATT&CK™ Matrices Tactics Techniques Mitigations Groups Software Resources Blog Contribute Search site

Mitigations

Mitigation	Description
Code Signing	Set PowerShell execution policy to execute only signed scripts.
Disable or Remove Feature or Program	It may be possible to remove PowerShell from systems when not needed, but a review should be performed to assess the impact to an environment, since it could be in use for many legitimate purposes and administrative functions. Disable/restrict the WinRM Service to help prevent uses of PowerShell for remote execution.
Privileged Account Management	When PowerShell is necessary, restrict PowerShell execution policy to administrators. Be aware that there are methods of bypassing the PowerShell execution policy, depending on environment configuration.

Mitigation Guidance From The Community

CERT-EU
FOR THE EU INSTITUTIONS, BODIES AND AGENCIES

CERT-EU Security Advisory 2019-021

Detecting and Preventing Emotet 2019 Campaign

September 30, 2019 — v1.0

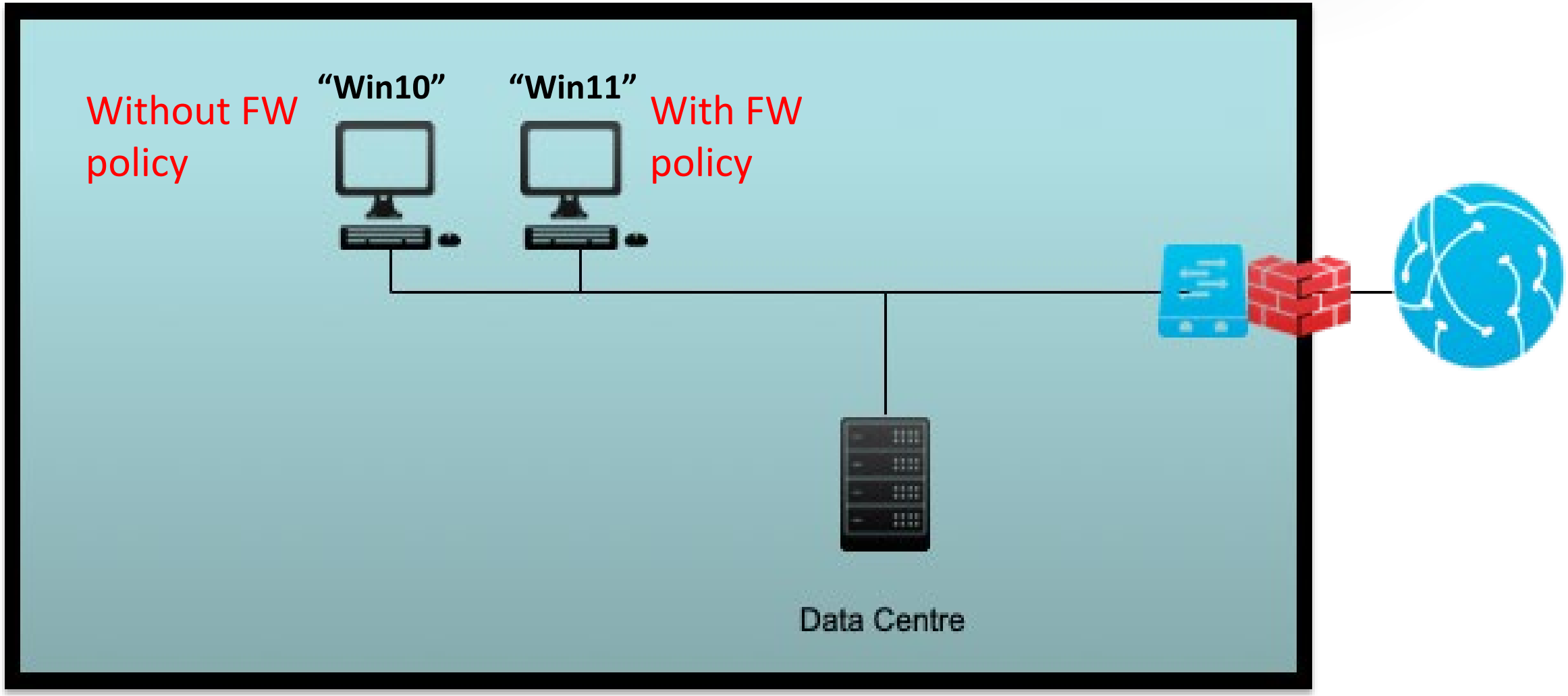
CERT-EU Security Whitepaper 2019-001

PowerShell – Cybersecurity Perspective

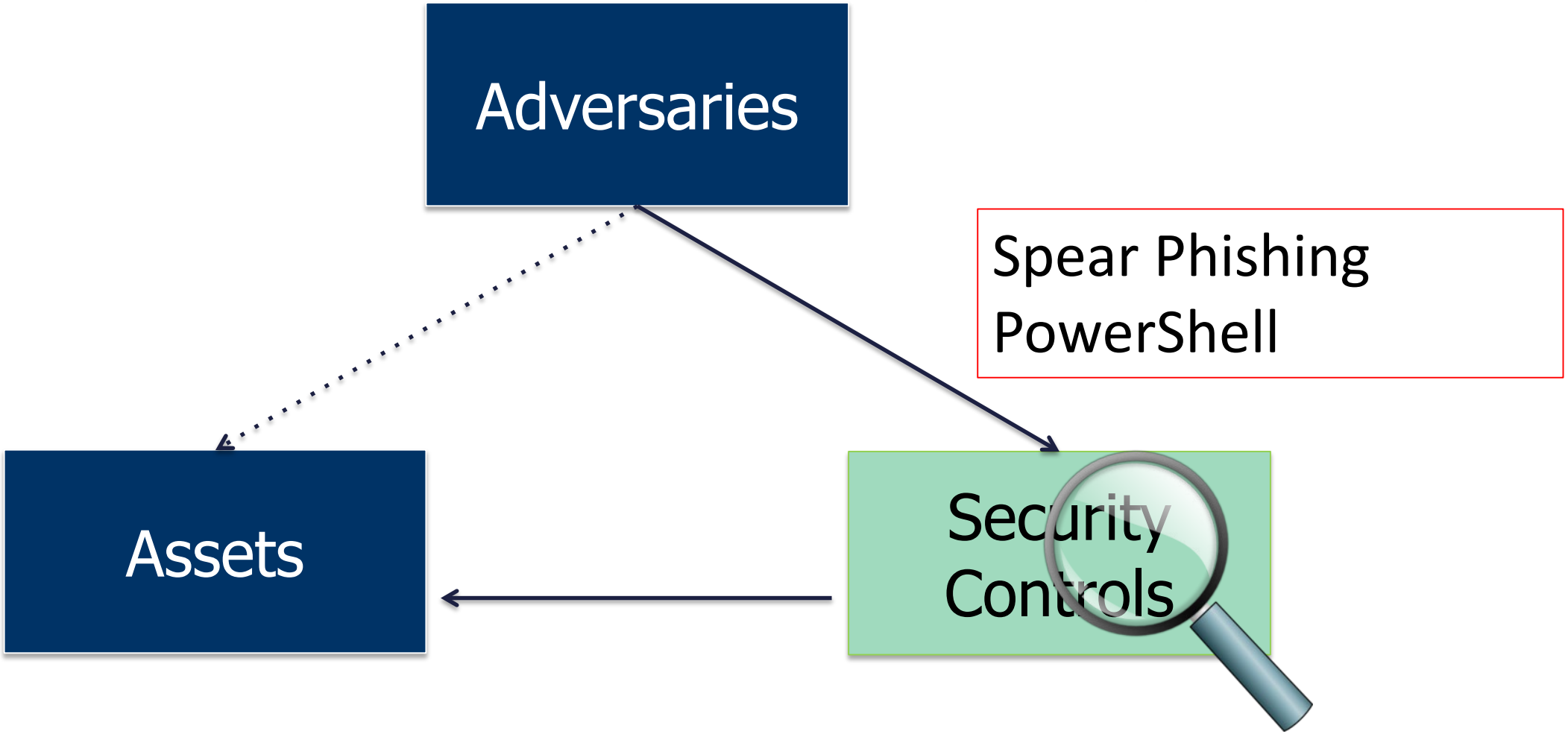
PREVENT Legitimate Windows Executables To Be Used To Gain Initial Foothold In Your Infrastructure

Dimitris Margaritis Follow
Nov 24 · 4 min read

Implemented In Our Enterprise Environment



Validate Our Controls In Our Lab



Screenshot of the lure document

win10.windomain.local (Win10_Office_Proxy) [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

test-document_2.docm - Word

DATEI START EINFÜGEN ENTWURF SEITENLAYOUT VERWEISE SENDUNGEN ÜBERPRÜFEN ANSICHT

SICHERHEITSWARNUNG Makros wurden deaktiviert. Inhalt aktivieren

Microsoft

Accept the license agreement

You can use Microsoft Word until Friday, September 20, 2019.
After that date, most features of Microsoft Word will be disabled.

To accept and start Word click Enable Editing and click Enable Content

SEITE 1 VON 1 0 WÖRTER DEUTSCH (DEUTSCHLAND) 100%

DEU 1:34 PM 12/18/2019

Right Ctrl

win11.windomain.local (Win11_Office_Proxy) [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

test-document_2.docm - Word

DATEI START EINFÜGEN ENTWURF SEITENLAYOUT VERWEISE SENDUNGEN ÜBERPRÜFEN ANSICHT

SICHERHEITSWARNUNG Makros wurden deaktiviert. Inhalt aktivieren

Microsoft

Accept the license agreement

You can use Microsoft Word until Friday, September 20, 2019.
After that date, most features of Microsoft Word will be disabled.

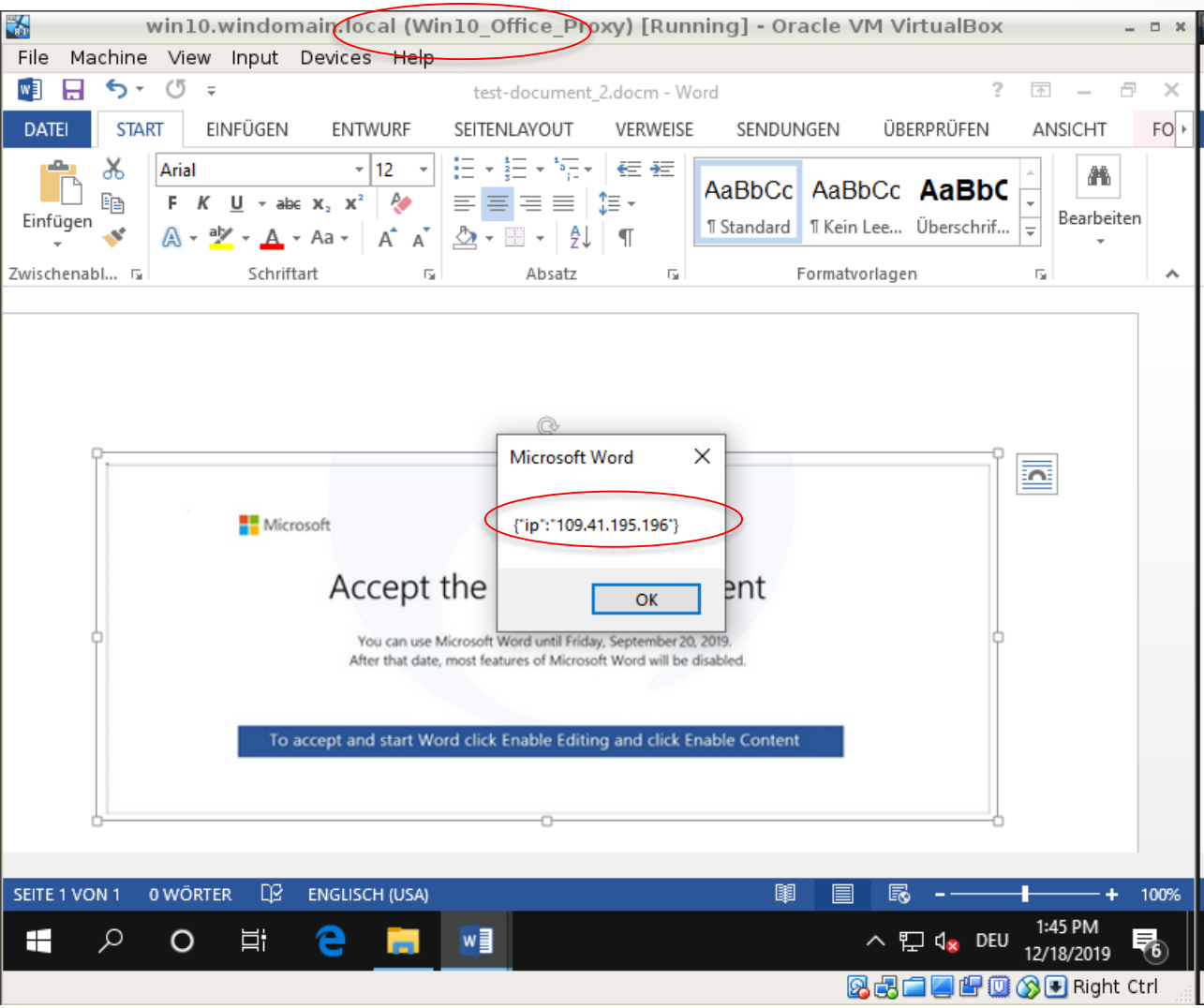
To accept and start Word click Enable Editing and click Enable Content

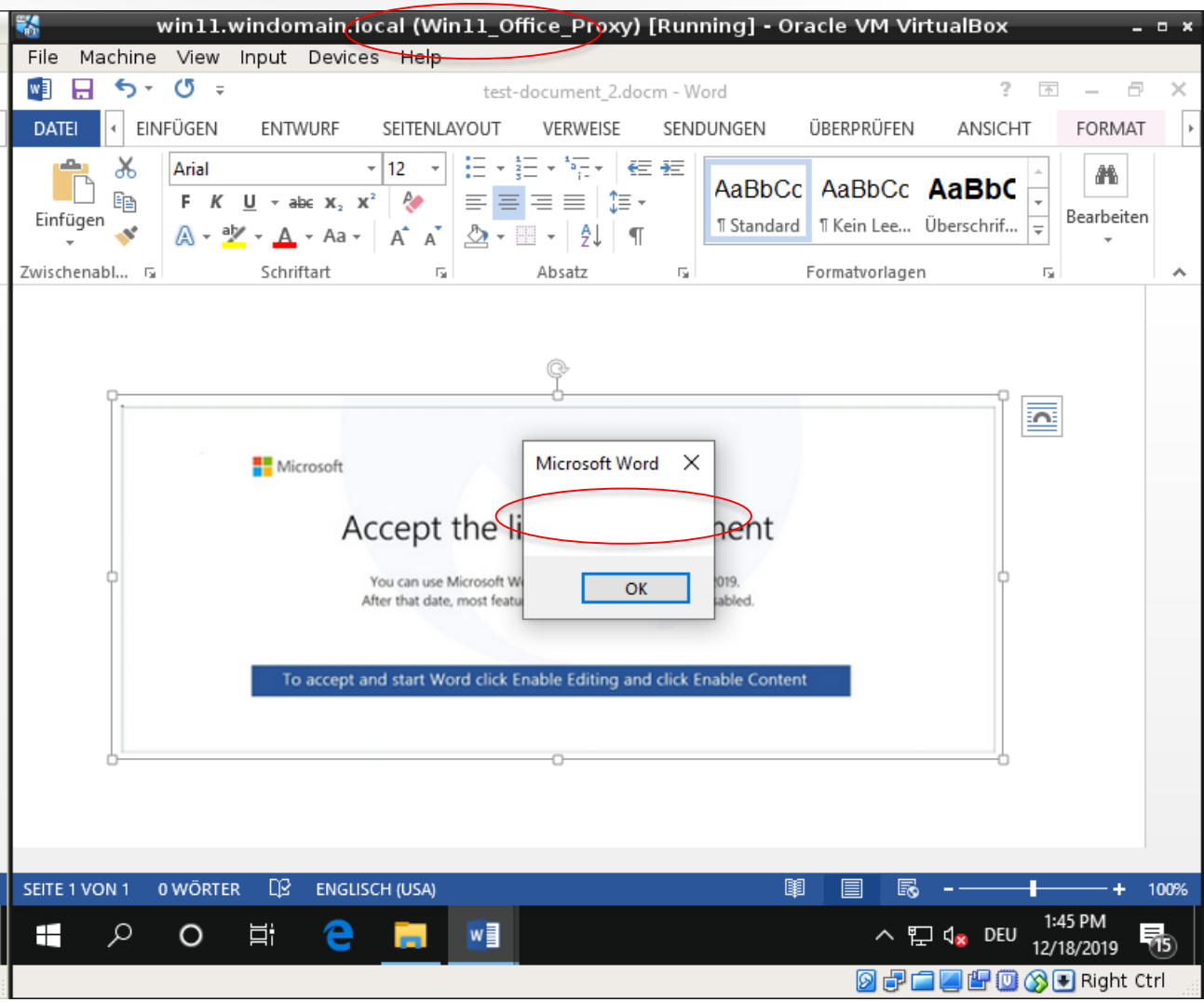
SEITE 1 VON 1 0 WÖRTER DEUTSCH (DEUTSCHLAND) 100%

DEU 1:34 PM 12/18/2019

Right Ctrl

Result Of The Execution Of The Macro





Visibility In Our Environment

- Screenshot in Splunk logs (Sysmon and proxy)

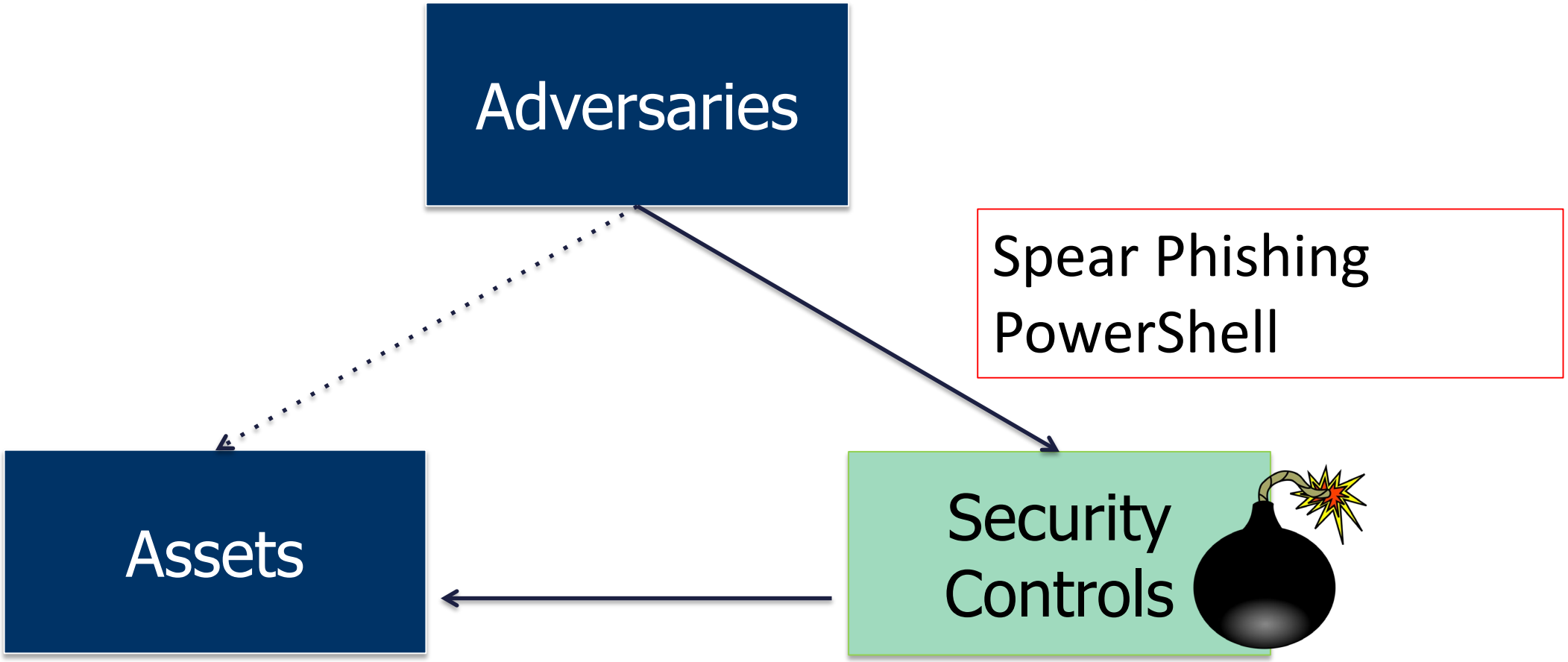
Events (3) Patterns **Statistics (3)** Visualization

20 Per Page ▾ Format Preview ▾

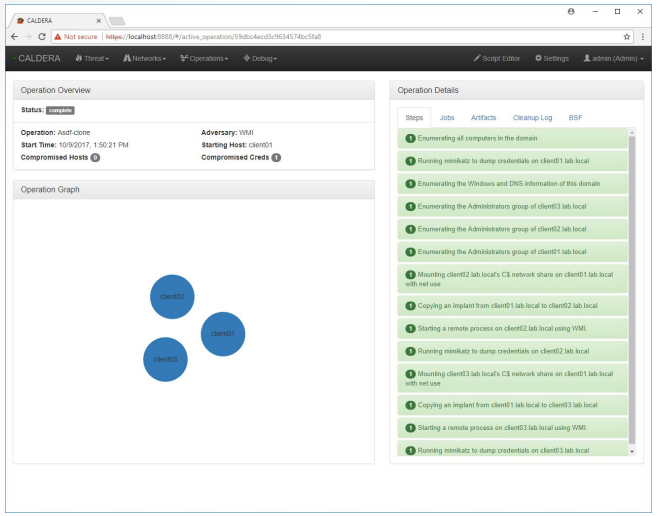
_time	source	ComputerName	src_ip	domain	CommandLine
2019-12-18 13:41:50.000	WinEventLog:Microsoft-Windows-Sysmon/Operational	win11.windomain.local			powershell -command "$Client = New-Object -TypeName System.Net.WebClient; $Client.Proxy.Credentials = [System.Net.CredentialCache]::DefaultNetworkCredentials; $Client.DownloadString('https://api.ipify.org?format=json')"
2019-12-18 13:42:55.000	WinEventLog:Microsoft-Windows-Sysmon/Operational	win10.windomain.local			powershell -command "$Client = New-Object -TypeName System.Net.WebClient; $Client.Proxy.Credentials = [System.Net.CredentialCache]::DefaultNetworkCredentials; $Client.DownloadString('https://api.ipify.org?format=json')"
2019-12-18 13:43:28.141	/var/log/squid/access.log		192.168.38.104	api.ipify.org	

"Win10" (without FW rule)

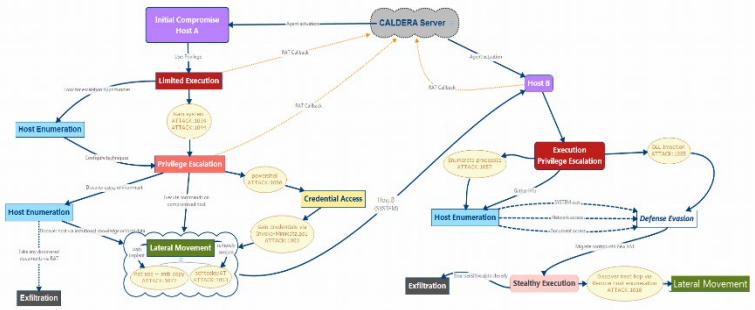
Test Our Controls



CALDERA – MITRE Open Source Research Project



- Automated adversary emulation
  - Safely replicate realistic adversary behavior
  - Repeatable testing and verification of prevention/detection
- Features
  - Uses ATT&CK to create Adversary profiles
  - Uses AI and modeling to make decisions about actions
  - Self-cleans after operation completes
  - Low install overhead
  - Does not require extensive red team knowledge to operate



Outcome Of Caldera With T1086 In Our Infrastructure

name	adversary	group	steps	planner
Powershell Execution	**Powershell Execution**	**my_group**	**14**	Powershell Execution collected 6 facts and used them to make decisions
The operation lasted (not finished yet)) with a random 4/8 second pause between steps	All Powershell Exections	2 agents were included	Powershell Execution was 78% successful in the attack	

att&ck

worked / failed	Tactic	Technique ID	Technique name
4 / 0	collection	T1086	PowerShell
5 / 3	execution	T1086	PowerShell
2 / 0	defense-evasion	T1086	PowerShell

Outcome On “Win11” (Protected With FW Policy)

2019-12-20 14:12:33	agent#126049... cmd.exe information gathering
2019-12-20 14:12:33	agent#126049... PowerShell information gathering
2019-12-20 14:12:33	agent#126049... Emulate Administrator Tasks
2019-12-20 14:12:33	agent#126049... Install PSTools
2019-12-20 14:12:33	agent#126049... PowerShell bitly Link Download
2019-12-20 14:12:33	agent#126049... PowerShell Invoke MimiKats
2019-12-20 14:12:33	agent#126049... Move Powershell & triage
2019-12-20 14:12:33	agent#10431... cmd.exe information gathering
2019-12-20 14:12:33	agent#10431... PowerShell information gathering
2019-12-20 14:12:33	agent#10431... Emulate Administrator Tasks
2019-12-20 14:12:33	agent#10431... Install PSTools
2019-12-20 14:12:33	agent#10431... PowerShell bitly Link Download
2019-12-20 14:12:33	agent#10431... PowerShell Invoke MimiKats
2019-12-20 14:12:33	agent#10431... Move Powershell & triage

RSA Conference 2020

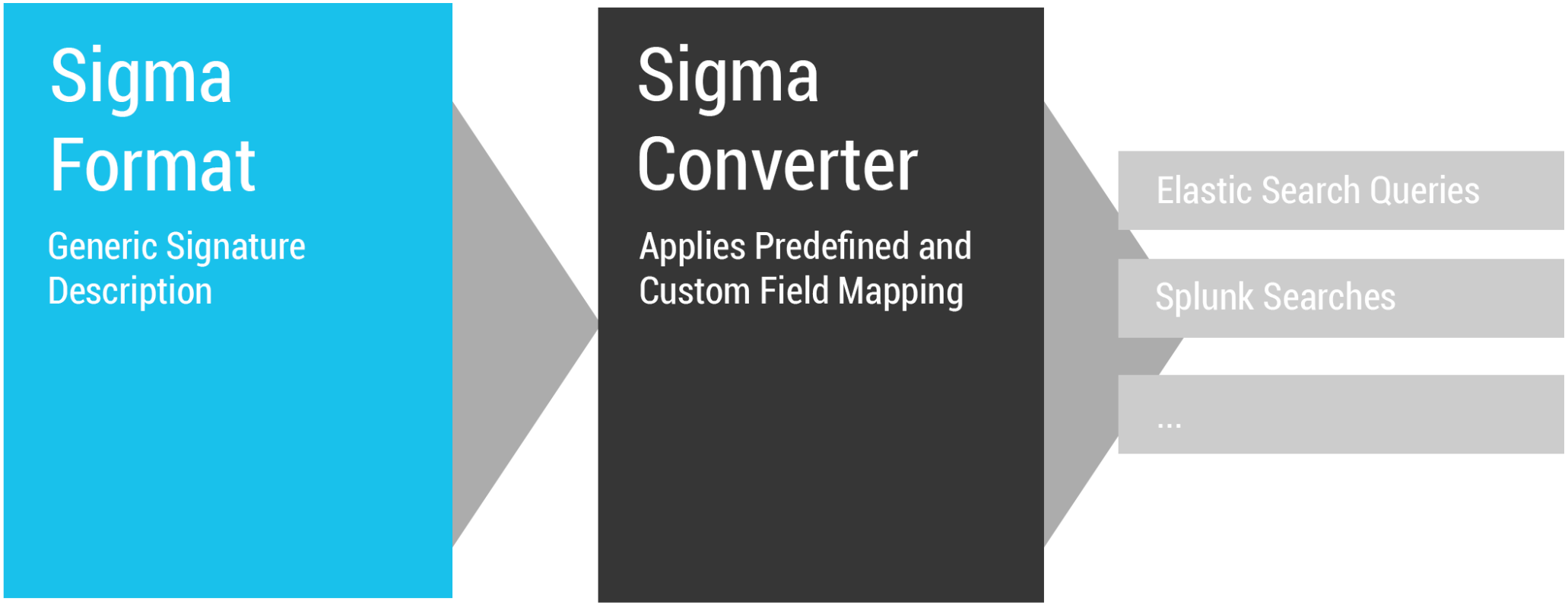
Detect

Design And Validate Our Analytics

Design Our Detection

- Gain Visibility
  - Priorities in log collection
- Design Analytics
  - Write them with knowledge of Our Adversaries
  - Get them from the community
- Deploy
  - Detect / Hunt / Refine

SIGMA: A Language for Analytics



https://github.com/Neo23x0/sigma

SIGMA Community Rules Repository

Branch: master ▾ sigma / rules / windows / powershell / Create new file | Find file | History

thomaspatzke Added UUIDs to rules — Latest commit 0592cbb 16 days ago

..		
powershell_data_compressed.yml	Added UUIDs to rules	16 days ago
powershell_downgrade_attack.yml	Added UUIDs to rules	16 days ago
powershell_exe_calling_ps.yml	Added UUIDs to rules	16 days ago
powershell_malicious_commandlets.yml	Added UUIDs to rules	16 days ago
powershell_malicious_keywords.yml	Added UUIDs to rules	16 days ago
powershell_ntfs_ads_access.yml	Added UUIDs to rules	16 days ago
powershell_prompt_credentials.yml	Added UUIDs to rules	16 days ago
powershell_psattack.yml	Added UUIDs to rules	16 days ago
powershell_shellcode_b64.yml	Added UUIDs to rules	16 days ago
powershell_suspicious_download.yml	Added UUIDs to rules	16 days ago
powershell_suspicious_invocation_generic.yml	Added UUIDs to rules	16 days ago
powershell_suspicious_invocation_specific.yml	Added UUIDs to rules	16 days ago
powershell_suspicious_keywords.yml	Added UUIDs to rules	16 days ago
powershell_winlogon_helper_dll.yml	Added UUIDs to rules	16 days ago

55 lines (55 sloc) | 1.73 KB

Raw | Blame | History

```yaml
title: Microsoft Office Product Spawning Windows Shell
id: 438025f9-5856-4663-83f7-52f878a70a50
status: experimental
description: Detects a Windows command line executable started from Microsoft Word, Excel, Powerpoint, Publisher and Visio.
references:
    - https://www.hybrid-analysis.com/sample/465aabe132ccb949e75b8ab9c5bda36d80cf2fd503d52b8bad54e295f28bbc21?environmentId=100
    - https://mgreen27.github.io/posts/2018/04/02/DownloadCradle.html
tags:
    - attack.execution
    - attack.defense_evasion
    - attack.t1059
    - attack.t1202
    - car.2013-02-003
    - car.2014-04-003
author: Michael Haag, Florian Roth, Markus Neis
date: 2018/04/06
logsource:
    category: process_creation
    product: windows
detection:
    selection:
        ParentImage:
            - '*\WINWORD.EXE'
            - '*\EXCEL.EXE'
            - '*\POWERPNT.exe'
            - '*\MSPUB.exe'
            - '*\VISIO.exe'
            - '*\OUTLOOK.EXE'
        Image:
            - '*\cmd.exe'
            - '*\powershell.exe'
            - '*\wscript.exe'
            - '*\cscript.exe'
            - '*\sh.exe'
            - '*\bash.exe'
            - '*\scrcons.exe'
            - '*\schtasks.exe'
            - '*\regsvr32.exe'
            - '*\hh.exe'
            - '*\wmic.exe'  # https://app.any.run/tasks/c903e9c8-0350-440c-8688-3881b556b8e0/
            - '*\mshta.exe'
            - '*\rundll32.exe'
            - '*\msiexec.exe'
            - '*\forfiles.exe'
            - '*\scriptrunner.exe'
            - '*\mftrace.exe'
            - '*\AppVLP.exe'
            - '*\svchost.exe'  # https://www.vmray.com/analyses/2d2fa29185ad/report/overview.html
    condition: selection
fields:
    - CommandLine
    - ParentCommandLine
falsepositives:
    - unknown
level: high
```


- Detecting Windows command line executable spawned from Microsoft Office

Detection With SIGMA Rules

- Splunk alerts detecting PowerShell spawned from Word

win_office_shell

Save | Save As | View | Close

```
index=* ((ParentImage="*\\WINWORD.EXE" OR ParentImage="*\\EXCEL.EXE" OR ParentImage="*\\POWERPNT.exe" OR ParentImage="*\\MSPUB.exe" OR ParentImage="*\\VISIO.exe" OR ParentImage="*\\OUTLOOK.EXE") (Image="*\\cmd.exe" OR Image="*\\powershell
    .exe" OR Image="*\\wscript.exe" OR Image="*\\cscript.exe" OR Image="*\\sh.exe" OR Image="*\\bash.exe" OR Image="*\\scrcons.exe" OR Image="*\\schtasks.exe" OR Image="*\\regsvr32.exe" OR Image="*\\hh.exe" OR Image="*\\wmic.exe" OR Image="
    *\\mshta.exe" OR Image="*\\rundll32.exe" OR Image="*\\msiexec.exe" OR Image="*\\forfiles.exe" OR Image="*\\scriptrunner.exe" OR Image="*\\mftrace.exe" OR Image="*\\AppVLP.exe" OR Image="*\\svchost.exe")) | table _time host CommandLine
    ,ParentCommandLine
```

Last 60 minutes

✓ 2 events (12/18/19 1:20:00.000 PM to 12/18/19 2:20:55.000 PM) | No Event Sampling | Job | Verbose Mode

Events (2) | Patterns | Statistics (2) | Visualization

20 Per Page | Format | Preview

_time	host	CommandLine	ParentCommandLine
2019-12-18 13:42:55	win10	powershell -command "$Client = New-Object -TypeName System.Net.WebClient; $Client.Proxy.Credentials = [System.Net.CredentialCache]::DefaultNetworkCredentials; $Client.DownloadString('https://api.ipify.org?format=json')"	"C:\Program Files (x86)\Microsoft Office\Office15\WINWORD.EXE" /n "C:\vagrant\resources\test-document_2.docm" /o ""
2019-12-18 13:41:50	win11	powershell -command "$Client = New-Object -TypeName System.Net.WebClient; $Client.Proxy.Credentials = [System.Net.CredentialCache]::DefaultNetworkCredentials; $Client.DownloadString('https://api.ipify.org?format=json')"	"C:\Program Files (x86)\Microsoft Office\Office15\WINWORD.EXE" /n "C:\vagrant\resources\test-document_2.docm" /o ""

Detection With SIGMA Rules (2)

- Splunk alert detecting PowerShell communicating outside

sysmon_powershell_network_connection

Save | Save As | View | Close

```
index=* (source="WinEventLog:Microsoft-Windows-Sysmon/Operational" (EventCode="3" Image="*\\powershell.exe" Initiated="true") NOT ((DestinationIp="10.*" OR DestinationIp="192.168.*" OR DestinationIp="172.16.*" OR DestinationIp="172.17.*" OR
    DestinationIp="172.18.*" OR DestinationIp="172.19.*" OR DestinationIp="172.20.*" OR DestinationIp="172.21.*" OR DestinationIp="172.22.*" OR DestinationIp="172.23.*" OR DestinationIp="172.24.*" OR DestinationIp="172.25.*" OR DestinationIp
    ="172.26.*" OR DestinationIp="172.27.*" OR DestinationIp="172.28.*" OR DestinationIp="172.29.*" OR DestinationIp="172.30.*" OR DestinationIp="172.31.*" OR DestinationIp="127.0.0.1") DestinationIsIpv6="false" User="NT AUTHORITY\\SYSTEM"))
    |table _time host DestinationIp
```

Last 60 minutes

✓ 1 event (12/18/19 1:30:00.000 PM to 12/18/19 2:30:36.000 PM) | No Event Sampling | Job | Verbose Mode

Events (1) | Patterns | Statistics (1) | Visualization

20 Per Page | Format | Preview

_time	host	DestinationIp
2019-12-18 13:43:32	win10	192.168.38.105

Alert on “Win10”
(without FW rule)

Detection With SIGMA Rules – Building Alerts (3)

- Splunk alerts built with identified SIGMA rules

	Time	Fired alerts	App	Type	Severity	Mode	Actions
☐	2019-12-18 14:42:06 UTC	sysmon_powershell_network_connection	search	Real-time	Critical	Per Result	View results | Edit search | Delete
☐	2019-12-18 14:41:34 UTC	win_office_shell	search	Real-time	Medium	Per Result	View results | Edit search | Delete
☐	2019-12-18 14:41:29 UTC	win_office_shell	search	Real-time	Medium	Per Result	View results | Edit search | Delete

Critical alert on “Win10”
(without FW rule)

Alerts Triggered By Running Caldera With T1086

	Time	Fired alerts	App	Type	Severity	Mode	Actions
☐	2019-12-20 13:19:50 UTC	sysmon_powershell_network_connection	search	Real-time	Critical	Per Result	View results | Edit search | Delete
☐	2019-12-20 13:19:41 UTC	sysmon_powershell_network_connection	search	Real-time	Critical	Per Result	View results | Edit search | Delete
☐	2019-12-20 13:19:40 UTC	sysmon_powershell_network_connection	search	Real-time	Critical	Per Result	View results | Edit search | Delete
☐	2019-12-20 13:19:39 UTC	sysmon_powershell_network_connection	search	Real-time	Critical	Per Result	View results | Edit search | Delete

All alerts are on “Win10”
(without FW rule)

RSAConference2020

Update

Update on ATT&CK Developments

- ATT&CK for ICS, Cloud and more
- Subtechniques
- Threat Report ATT&CK Mapper (TRAM)
- MITRE ENGENUITY

RSAConference2020

Share

Contribute To The Community

Share Insights And Contribute

- The MITRE ATT&CK community is very active
- Sharing TTPs/SIGMA rules is easier and more useful than IOCs
  - Contribute to MITRE ATT&CK attack@mitre.org
  - Contribute to SIGMA https://github.com/Neo23x0/sigma/tree/master/rules
- Participate in the Community
  - MITRE ATT&CKcon
  - EU ATT&CK User Community

EU ATT&CK User Community

- Mailing list: opt in ? -> email to info@circl.lu
- Next workshop in Brussels 18-19 May 2020
- The biggest ATT&CK event ever…

Workshop - EU ATT&CK Community

Next workshop - event for EU ATT&CK Community

"Apply" Slide

- Next week you should:
  - Consider Windows Firewall policy to mitigate PowerShell techniques
- In the first three months following this presentation you should:
  - Identify Your Adversaries
  - Identify and deploy at least three use cases in your organization
- Within six months you should:
  - Permeate your cyber defense using ATT&CK
  - Share your insights in the SIGMA community

Resources And Acknowledgements

- ATT&CK repository and ATT&CK Navigator
- How to use the MITRE ATT&CK Navigator
- PREVENT Legitimate Windows Executables To Be Used To Gain Initial Foothold In Your Infrastructure (@dmargaritis)
- SIGMA and SIGMA rule collection (Thomas Patzke, Florian Roth)
- CALDERA
- EU ATT&CK Community Workshop 18-19 May 2020
- Munich Airport Information Security Hub
- Center for Threat-Informed Defense
- Detection Lab