

MENT

ments

L. Foot.

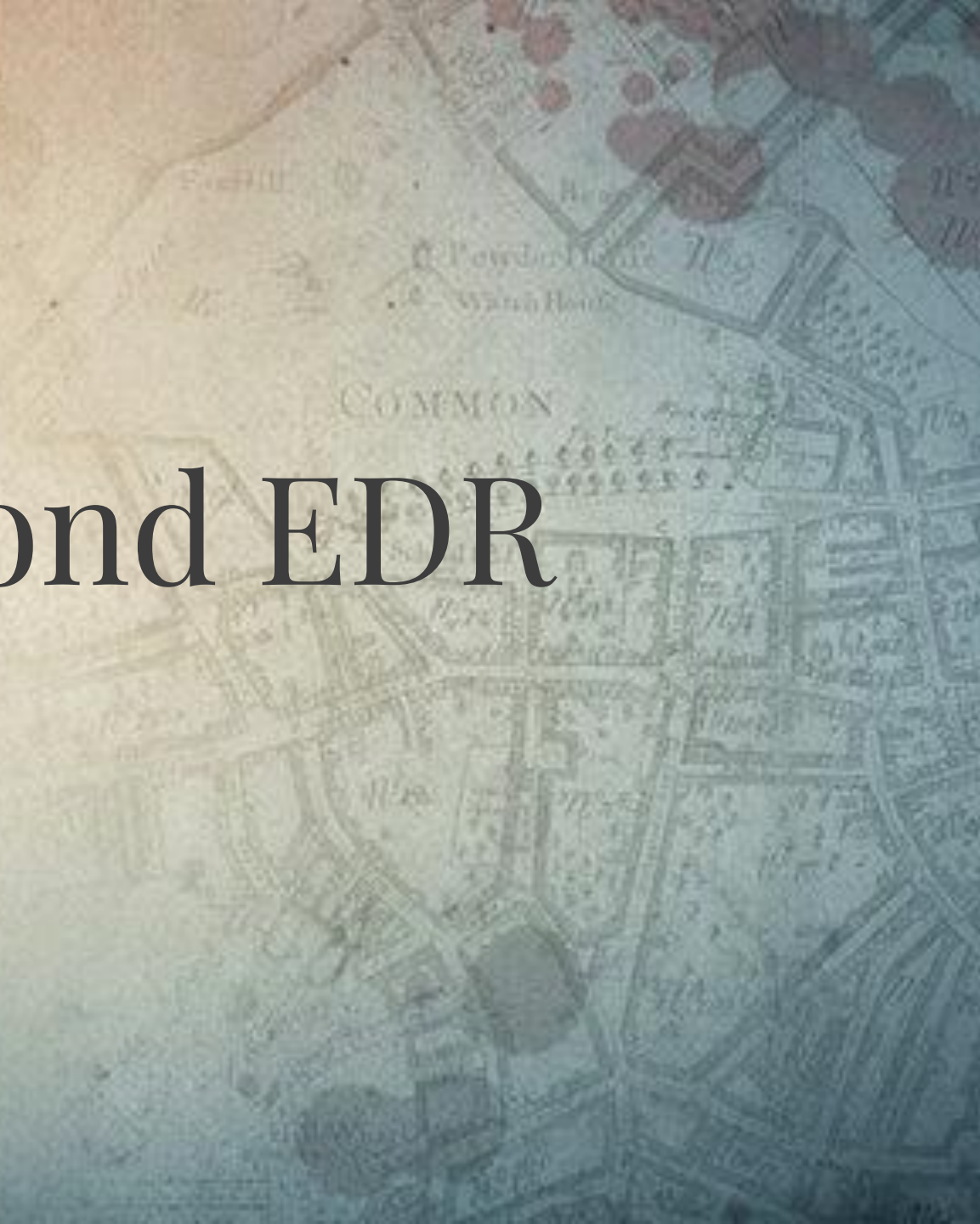
C. F.

Ld. F.

Fore A.



Visibility Beyond EDR



Intros

Greg Stachura

- Senior Manager @ Security Risk Advisors
- 20+ years in blue team operations
- Specializing in security operations, blue team tools, forensics and incident response
- Certifications:
 - GCFA
 - CISSP
 - Azure Administrator Associate
 - CompTIA CySA+

Alexandra Ioannidis

- Senior Consultant @ Security Risk Advisors
- 6+ years in blue team operations
- Specializing in EDR engineering, threat hunting and incident response
- Education
 - Graduate of RIT with a degree in Computing Security
 - CompTIA CySA+

Disclaimer

- We are not recreating EDR
- Tip of the iceberg
- Your mileage may vary based on:
 - EDR capabilities
 - Current security posture (crawl,walk,run)
 - Tools available

Why are we here?

- EDRs have varying levels of visibility
 - Continue to get better, but there is still room to grow
- Visibility vs detection engineering
- Native logging missing, obscured

Solutions?

- Add additional EDR?
- Replace your EDR?
- Add additional logging and collect everything?
- Add additional logging and collect what we need?

What do we need?

- Organizational dependent
 - Past incidents
 - Pen tests
 - Industry-specific Intel
- Regulations
- Purple teaming
- Threat hunting

How do we get more visibility?

- Native Events
 - Windows logging
 - Turn on more extensive logging
- Sysmon
- On-Demand forensics
- Get creative...

How to not break the bank?



Combined Pipeline



Pipeline at the Destination



Log Sources



**CRIBL
/Rsyslog/Tenzir**

Remove Noise
Remove Size
Route Intelligently



Data Lake



SIEM

Your Environment



Pipeline at the Source



Logging Options

Filter events
Route events
Partial/On-Demand
Events



Endpoints



Data Lake

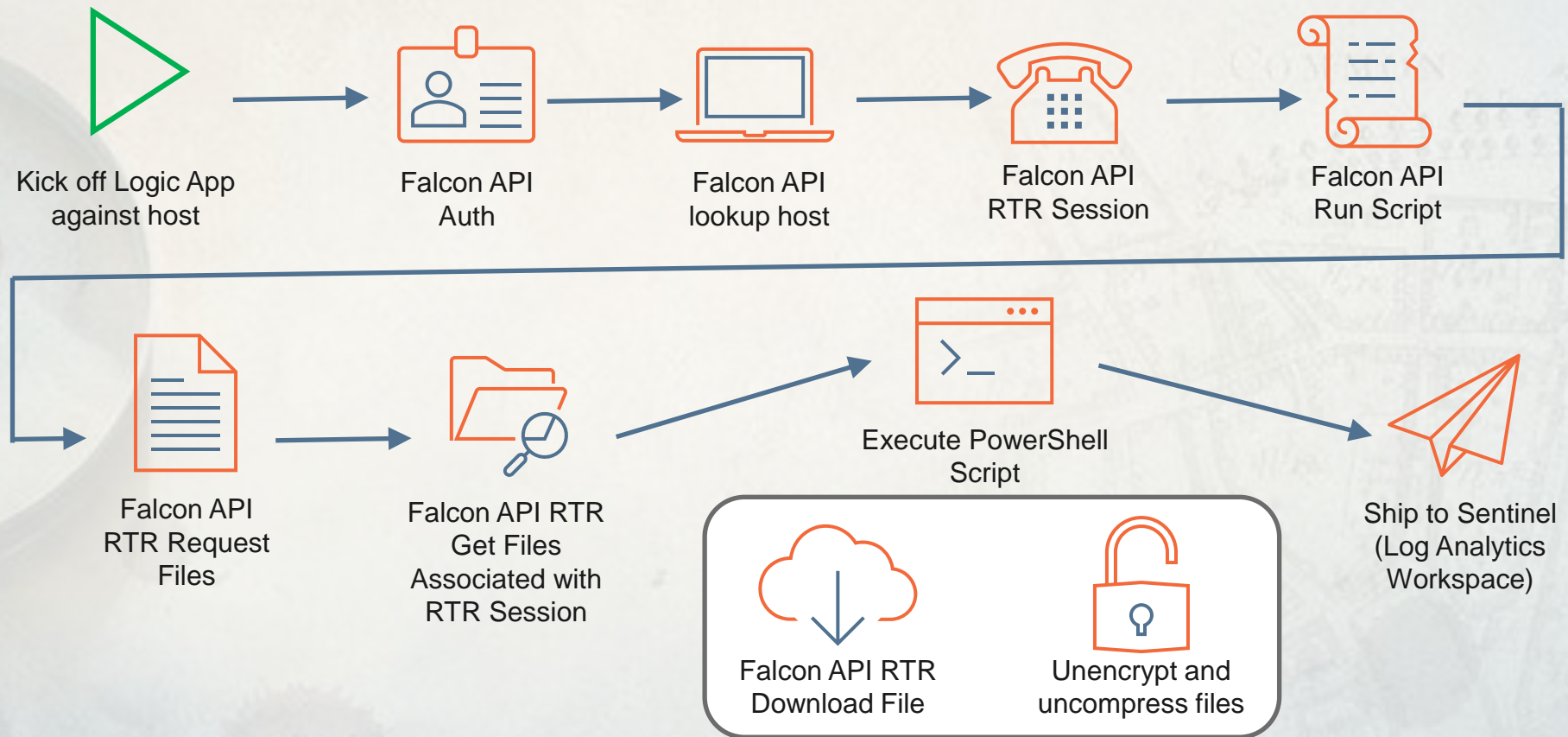


SIEM

On-Demand Forensics Example

- Pulling detailed logs when a potential incident happens
 - Keeps logging volume down
 - Logs are still being gathered, just not shipped
 - Not all positives to this approach...
- Run a script grab previously generated logs, and/or other important data
 - PS-Remoting
 - PSEXec
 - EDR Command Line access!
- Ship logs back to your SIEM!

On-Demand Forensics Example – CrowdStrike to Sentinel



PowerShell
Script Block

Process
Tampering

IMPHASH

Scheduled
Tasks

Service
Creation

EDR
Deficiency

Example EDR Gaps

- PowerShell Script Block Logging
- Process Tampering
- IMPHASH
- Scheduled Task
- Service Creation
- EDR Deficiencies
- Log Tampering

PowerShell Script Block Logging

Case Description:

- Some EDRs do not log any PowerShell script content
- MDE will partially log content
- CrowdStrike will not log script content, but will log commands run in the terminal
 - To log terminal commands, you will need to have the "Script-based Execution Visibility" prevention policy setting checked
 - Unable to develop custom IOAs on this data

Process
Tampering

IMPHASH

IMPHASH

Service
CreationEDR
Deficiency

Case File: PowerShell Script Block Logging**Evidence Item:** Testing Script**Evidence Description:**

```
# Generate a random number between 1 and 100
$randomNumber = Get-Random -Minimum 1 -Maximum 101

# Generate a random string of 8 characters
$randomString = -join ((65..90) + (97..122) | Get-
Random -Count 8 | % {[char]$_})

# Output the random values
Write-Output "Random number: $randomNumber"
Write-Output "Random string: $randomString"
```

Process
Tampering

IMPHASH

Scheduled
TasksService
CreationEDR
Deficiency

Case File: PowerShell Script Block Logging

Evidence Item: Microsoft Defender for Endpoint

Evidence Description:

- Discrepancies with logging this information
- Observed instances where all content was logged and other instances where only partial content was shown
- Was unable to see the name of the script or file path in which the script was located in these events

```
1 DeviceEvents
2 | where DeviceName contains "rss"
3 | where ActionType contains "PowerShell"
```

Getting started **Results** Query history

Export Show empty columns 5 items Search 00:00.154

Filters: Add filter

...	InitiatingProcessParentCrea...	InitiatingProcessLogonId	ReportId	AdditionalFields
Sep 30, 2024 3:35:57 PM	0	2744	["Command": "PSConsoleHostReadLine"]	
Sep 30, 2024 3:35:57 PM	0	2743	["Command": "prompt"]	
Sep 30, 2024 3:35:57 PM	0	2803	["Command": "cd .\\Desktop\\"]	
Sep 30, 2024 3:35:57 PM	0	2828	["Command": "\$randomNumber = Get-Random -Minimum 1 -Maximum 101"]	
Sep 30, 2024 3:35:57 PM	0	2829	["Command": "\$randomString = -join ((65..90) + (97..122) Get-Random -Count 8 % {[char]\$_})"]	

Process
Tampering

IMPHASH

Scheduled
TasksService
CreationEDR
Deficiency

Case File: PowerShell Script Block Logging

Evidence Item: Native Windows Logging

Evidence Description:

- Will grab the contents of the entire script block
- Includes name and file path of the script

Level	Date and Time	Source	ID	Task Category
Verbose	9/30/2024 7:33:48 PM	PowerShell (Micr...	4104	Execute a Remote ...
Verbose	9/30/2024 7:33:48 PM	PowerShell (Micr...	4104	Execute a Remote ...
Verbose	9/30/2024 7:33:48 PM	PowerShell (Micr...	4104	Execute a Remote ...

Event 4104, PowerShell (Microsoft-Windows-PowerShell)

General Details

```
Creating Scriptblock text (1 of 1):  
# Generate a random number between 1 and 100  
$randomNumber = Get-Random -Minimum 1 -Maximum 101  
  
# Generate a random string of 8 characters  
$randomString = -join ((65..90) + (97..122) | Get-Random -Count 8 | % {[char]$_})  
  
# Output the random values  
Write-Output "Random number: $randomNumber"  
Write-Output "Random string: $randomString"  
  
ScriptBlock ID: 0cd81bd3-a707-4f41-a236-315e85f11e41  
Path: C:\Users\ai-admin\Desktop\test.ps1
```

Log Name: Microsoft-Windows-PowerShell/Operational
Source: PowerShell (Microsoft-Windows-PowerShell) Logged: 9/30/2024 7:33:48 PM
Event ID: 4104 Task Category: Execute a Remote Command
Level: Verbose Keywords: None
User: rrs-testmachine\ai-admin Computer: rrs-testmachine
OpCode: On create calls
More Information: [Event Log Online Help](#)

Process
Tampering

IMPHASH

Scheduled
TasksService
CreationEDR
Deficiency

Process Tampering

Case Description:

- Involves manipulating process memory
- Includes techniques such as process hollowing and process herpaderping
- End goals are privilege escalation and defense evasion

IMPHASH

Scheduled
TasksService
CreationEDR
Deficiency

Case File: Process Tampering

Evidence Item: Sysmon

Evidence Description:

- Sysmon Event ID 25 – Process Image Change
- Detects process hollowing and process herpaderping
- Able to use ProcessGuid to correlate these events with process creation events to gather additional context

Level	Date and Time	Source	Even...	Task Category
Information	11/17/2020 10:30:35 AM	Sysmon	1	Process Create (rule: ProcessCreate)
Information	11/17/2020 10:30:35 AM	Sysmon	25	Process Tampering (rule: ProcessTampering)
Information	11/17/2020 10:30:34 AM	Sysmon	1	Process Create (rule: ProcessCreate)
Information	11/17/2020 10:30:34 AM	Sysmon	1	Process Create (rule: ProcessCreate)
Information	11/17/2020 10:30:08 AM	Sysmon	1	Process Create (rule: ProcessCreate)
Information	11/17/2020 10:30:08 AM	Sysmon	25	Process Tampering (rule: ProcessTampering)
Information	11/17/2020 10:30:07 AM	Sysmon	1	Process Create (rule: ProcessCreate)

Event 25, Sysmon

General Details

Process Tampering:
RuleName: -
UtcTime: 2020-11-17 18:30:34.733
ProcessGuid: {bee51ebb-16ca-5fb4-a202-000000005000}
ProcessId: 4212
Image: C:\Windows\SysWOW64\svchost.exe
Type: Image is replaced

Case File: Process Tampering

Evidence Item: Microsoft Defender for Endpoint/CrowdStrike Falcon

Evidence Description:

- Tested with ProcessHollowing.exe
- MDE and CrowdStrike detected the test binary being malicious but not the behavior
- MDE does not expose telemetry for this apart from the NtProtectVirtualMemoryApiCall action

Anomalous memory allocation in cmd.exe process memory

Event info

Event
Anomalous memory allocation in cmd.exe process memory

Event time: Oct 4, 2024 2:37:15 PM Action type: NtProtectVirtualMemoryApiCall

User
rss-testmachine\ai-admin

Entities
cmd.exe > ProcessHollowing.exe > cmd.exe

IMPHASH

Case Description:

- Import Hash (ImpHash) is a hashing method for PE executable files
 - Determined based on the file's import table
- Used for malware analysis and correlation
- Can solve the problem of attackers updating files which changes other hash values (ex. SHA256)
- Useful for threat intel and hunting for malicious binaries

Case File: IMPHash

Evidence Item: Limitations

Evidence Description:

- Only works on PE files
 - Will not work on PDFs, Microsoft Office files, etc.
- Limited to Windows
- False positives and negatives
- ImpHash Manipulation

Case File: IMPHash

Evidence Item: Sysmon

Evidence Description:

- Tested with Keylogger
- Event ID 1 lists MD5, SHA256, and IMPHASH

```
Event 1, Sysmon
General Details
Process Create:
RuleName: -
UtcTime: 2024-10-07 18:39:16.534
ProcessGuid: {f312bb63-2ad4-6704-ee2d-000000001000}
ProcessId: 3984
Image: C:\Users\ai-admin\Desktop\key_n\key_n.exe
FileVersion: 0.0.0.0
Description:
Product: -
Company: -
OriginalFileName: key_n.exe
CommandLine: key_n.exe
CurrentDirectory: C:\Users\ai-admin\Desktop\key_n\
User: rss-testmachine\ai-admin
LogonGuid: {f312bb63-3c04-6700-bed7-160000000000}
LogonId: 0x16D7BE
TerminalSessionId: 3
IntegrityLevel: Medium
Hashes: MD5=3D6F6CE009AD7CB777D63586B825DA77,SHA256=0BB0639E51BFA7EA2AD5ADE7909563787C089189BF5EDF295128C535FDAEB1A9,IMPHASH=F34D5F2D4577ED6D9CEEC516C1F5A744
ParentProcessGuid: {f312bb63-2ad0-6704-ea2d-000000001000}
ParentProcessId: 1144
ParentImage: C:\Windows\System32\cmd.exe
ParentCommandLine: "C:\Windows\System32\cmd.exe"
ParentUser: rss-testmachine\ai-admin
```


Case File: IMPHash

Evidence Item: Example

Evidence Description:

- SHA256 of the keylogger did not return any results on VirusTotal but the IMPHASH does

0BB0639E51BFA7EA2AD5ADE7909563787C089189BF5EDF295128C535FDAEB1A9



No matches found

Alternatively, do you want to locate your threat based on static, dynamic, content, attribution or other advanced IoC context? Our platform allows you to search across our entire threat corpus using a [myriad of modifiers](#), [learn more](#).

[Try out our offering](#) [Try a new search](#)

MALWARE bazaar by ABUSE™ [Browse](#) [Upload](#) [Hunting](#) [API](#) [Export](#) [Statistics](#)

Browse Database

imphash:F34D5F2D4577ED6D9CEEC516C1F5A744

Search Syntax [?](#)

Date (UTC)	SHA256 hash	Type	Signature	Tags
2024-10-04 14:40	32058aa91a7e956ae9b4...	exe	DCRat	DCRat exe
2024-10-04 11:15	39c548d4cf4fdb68e52b...	exe	DCRat	DCRat exe
2024-10-04 09:31	bb0db766edcbcd8852b6...	exe	MassLogger	exe MassLogger
2024-10-04 09:25	5b82fc5190c0d6644a7ea...	exe	Loki	exe Loki
2024-10-04 08:59	53a5dd5f5446a75a32253...	exe	Formbook	exe FormBook
2024-10-04 08:54	0ad205b2d883bca56250...	exe	AgentTesla	AgentTesla exe
2024-10-04 08:53	3376b495c19dc3e179dfb...	exe	SnakeKeylogger	exe SnakeKeylogger
2024-10-04 08:52	4ed81a9a25e52a99d768...	exe	RemcosRAT	exe pif RemcosRAT
2024-10-04 08:51	5a11fb6ef4be30e7c7a65...	exe	Formbook	exe FormBook
2024-10-04 08:50	b9f4537fa4b470f09cc62c...	exe	SnakeKeylogger	exe SnakeKeylogger
2024-10-04 08:48	780354ea81fa066b08576...	exe	SnakeKeylogger	exe SnakeKeylogger

Scheduled Tasks

Case Description:

- Persistence mechanism used by attackers
- Can be accomplished via command-line or GUI
- Evidence of GUI scheduled tasks looks different than command-line creation

PowerShell
Script Block

Process
Tampering

IMPHASH

Scheduled
Tasks

Service
Creation

EDR
Deficiency

Case File: Scheduled Tasks**Evidence Item:** CrowdStrike**Evidence Description:**

- CrowdStrike will log the creation of new scheduled tasks when done via PowerShell commands or via the GUI
 - #event_simpleName: ScheduledTaskRegistered
- This telemetry data is not alerted on natively and cannot be accessed via the custom IOAs to create a detection rule

Case File: Scheduled Tasks

Evidence Item: Native Windows Logging

Evidence Description:

- Gap can be addressed with windows event id 4698
 - Will be able to see all scheduled tasks being created no matter what method they were added
 - Can use this data to create detections

Service Creation

Case Description:

- Persistence mechanism used by attackers
- Unlike Scheduled Tasks there is no native GUI creation tool
- Use of sc.exe (command prompt) or New-Service (PowerShell) can be used to create Services

PowerShell
Script Block

Process
Tampering

IMPHASH

Scheduled
Tasks

Service
Creation

Case File: Service Creation**Evidence Item:** Microsoft Defender for Endpoint**Evidence Description:**

- Microsoft Defender for Endpoint logs potentially new services with:
 - ActionType = ServiceInstalled
- Overly verbose, thus false positive prone

Case File: Service Creation**Evidence Item:** CrowdStrike Falcon**Evidence Description:**

- No specific event created in CrowdStrike to highlight this type of activity
- Requires keying off command line of a process event to catch this activity

Case File: Service Creation**Evidence Item:** Native Windows Logging**Evidence Description:**

- Gap can be addressed with windows event ID 7045 (System event)
 - Will be able to see all new service creation events being created no matter what method they were added
 - Can use this data to create detections

EDR Deficiencies

Case Description:

- EDR can be bypassed or tampered with to inhibit it from working as expected
- May involve killing security software processes, modifying the tools so that they do not operate properly, etc.
- Can be achieved through stopping services, PowerShell cmdlets, via the Registry, or other methods

PowerShell
Script Block

Process
Tampering

IMPHASH

Scheduled
Tasks

Service
Creation

EDR
Deficiency

Log
Tampering

Case File: EDR Bypass

Evidence Item: Microsoft Defender for Endpoint and Windows Firewall

Evidence Description:

- Microsoft-Windows-Windows Defender/Operational
 - Event ID 5001 signals the disabling of Defender's Real-Time Protection
 - Event ID 5013 signals when Defender setting changes were blocked
- Windows Event code 7036 from the System log identifies if a service has stopped or started
 - MITRE: (source="WinEventLog:System" EventCode="7036")
ServiceName="Windows Defender" OR
ServiceName="Windows Firewall" AND
ServiceName="stopped*"

Log Tampering

Case Description:

- Defense evasion technique to avoid detection and obstruct investigations
- Includes modifying, falsifying, or deleting logs
- Can be cleared by various means including PowerShell, Wevtutil, and the event viewer GUI

Case File: Event Log Clearing/Disabling

Evidence Item: Microsoft Windows

Evidence Description:

- Clearing Logs
 - Example: `wevtutil cl Security, Clear-EventLog, Remove-EventLog`
 - Detected by: Security Event ID 1102 or System Event ID 104
 - Will also be able to use Event ID 4104 to monitor for this activity in PowerShell script blocks
- Disable the Event Log Service
 - Example: `sc stop EventLog`
 - Detected by: Service Control Manager Event ID 7035

A wooden desk with a magnifying glass, binoculars, a compass, and some money. The magnifying glass is in the bottom left, binoculars are in the middle left, a compass is in the top left, and some money is in the top left. A piece of white paper with text is in the center right, and a pencil is on the right side of the paper.

Conclusion

- Most EDRs have differing levels of visibility
- Sysmon and native Windows event logging can help augment gaps observed with EDR
- More robust logging can also provide additional telemetry data that may be useful during investigations and incident response



Questions?

A top-down view of a wooden desk. On the left, there is a stack of Indonesian Rupiah banknotes (10,000 and 100,000 denominations), a small round compass, and a black camera. In the center, a white piece of paper is placed on the desk. On the right, a pencil lies vertically. In the bottom left, a magnifying glass is partially visible.

Contact Us

Gregory.Stachura@sra.io

Alexandra.Ioannidis@sra.io

Preso Info (coming soon):

<https://github.com/theowlery/RSS-2024>