Sigma

Generic Signatures for Log Events

Thomas Patzke, 18. October 2017

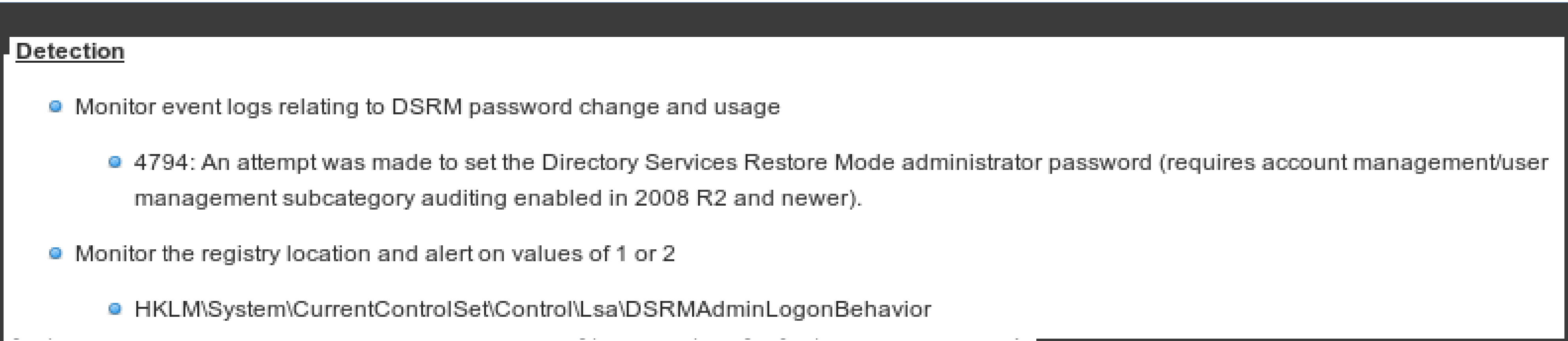
Agenda

- Threat Detection by Log Analysis
- Problems and Motivation
- Sigma – The Open Source Approach
  - Rule Format
  - Rule Examples
  - Conversion to SIEM queries
- Community and Contributors
- Current State and Future Plans

Threat Detection with Log Monitoring

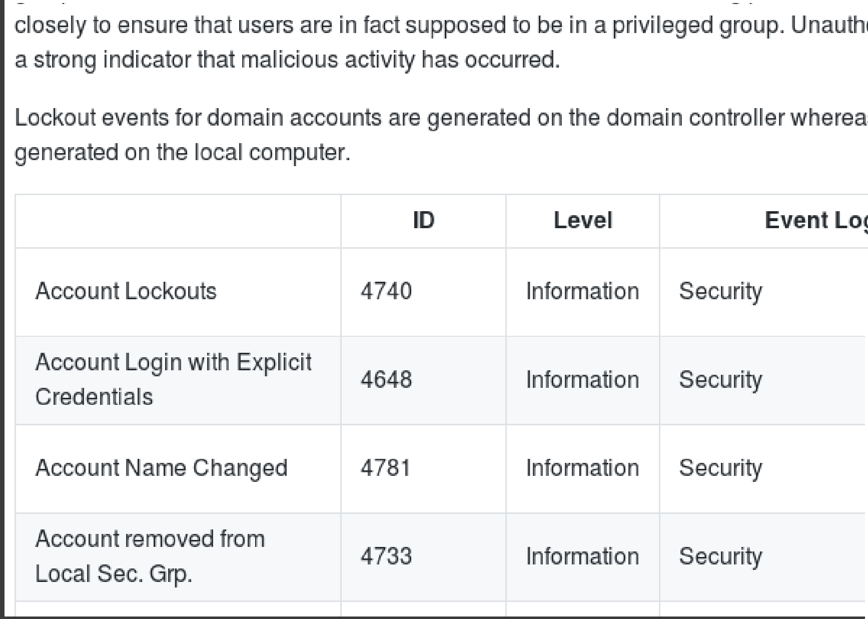
- Authentication & Accounts:
  - Large number of failed logon attempts
  - Alternation and usage of specific accounts (e.g. DSRM)
  - SID history
- Process Execution:
  - Execution from unusual locations
  - Suspicious process relationships
  - Known executables with unknown hashes
  - Known evil hashes
- Windows Events:
  - Service installations with rare names in monitored environment
  - New domain trusts
- Network: Port Scans, Host Discovery (Ping Sweeps)
- (Web) Applications: 5xx Errors, specific exceptions

Problems?



Detection

- Monitor event logs relating to DSRM password change and usage
  - 4794: An attempt was made to set the Directory Services Restore Mode administrator password (requires account management/user management subcategory auditing enabled in 2008 R2 and newer).
- Monitor the registry location and alert on values of 1 or 2
  - HKLM\System\CurrentControlSet\Control\Lsa\DSRMAdminLogonBehavior



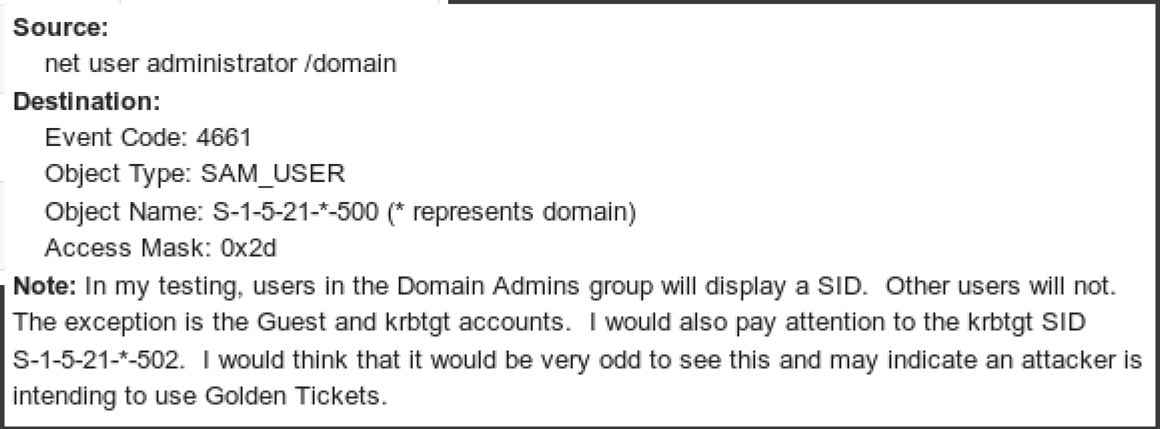
closely to ensure that users are in fact supposed to be in a privileged group. Unauthorized membership in privileged groups is a strong indicator that malicious activity has occurred.

Lockout events for domain accounts are generated on the domain controller whereas lockout events for local accounts are generated on the local computer.

	ID	Level	Event Log	Event Source
Account Lockouts	4740	Information	Security	Microsoft-Windows-Security-Auditing
Account Login with Explicit Credentials	4648	Information	Security	
Account Name Changed	4781	Information	Security	
Account removed from Local Sec. Grp.	4733	Information	Security	

Source: https://adsecurity.org/?p=1714

Source: https://findingbad.blogspot.de/2017/01/hunting-what-does-it-look-like.html



Source:
net user administrator /domain

Destination:
Event Code: 4661
Object Type: SAM_USER
Object Name: S-1-5-21-*-500 (* represents domain)
Access Mask: 0x2d

Note: In my testing, users in the Domain Admins group will display a SID. Other users will not. The exception is the Guest and krbtgt accounts. I would also pay attention to the krbtgt SID S-1-5-21-*-502. I would think that it would be very odd to see this and may indicate an attacker is intending to use Golden Tickets.

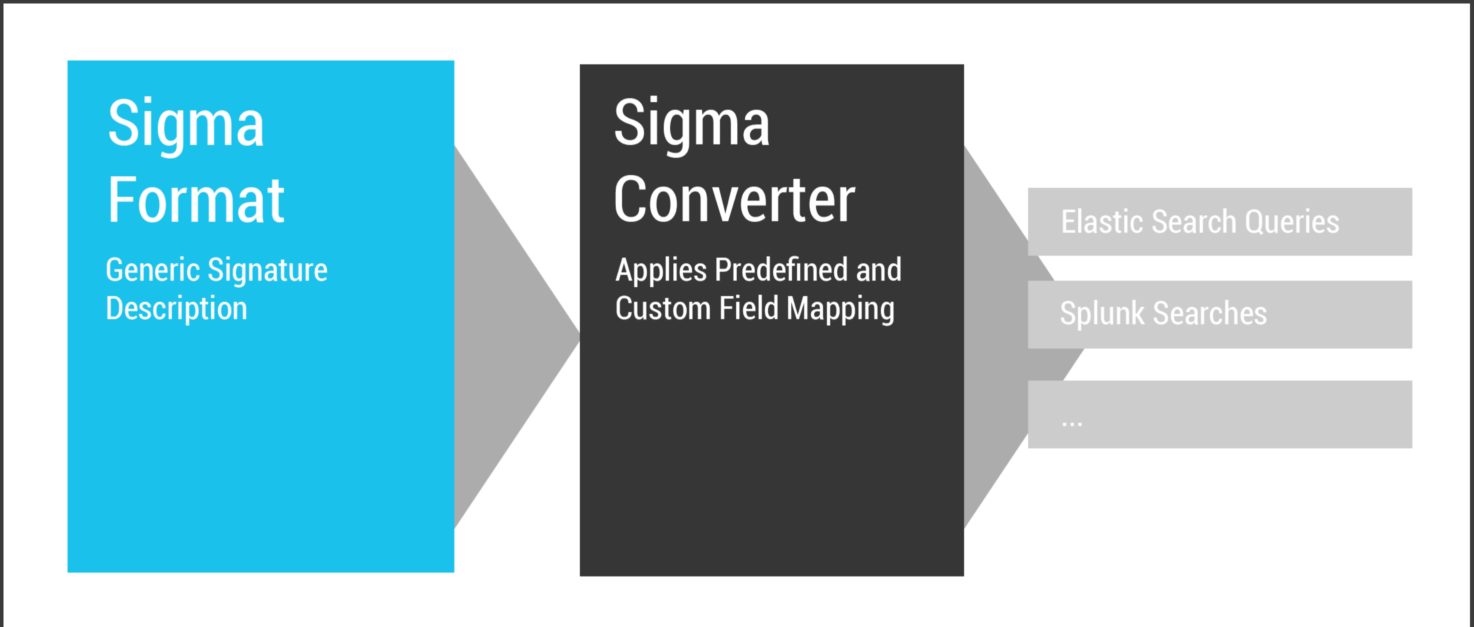
Source: https://github.com/iadgov/Event-Forwarding-Guidance/tree/master/Events

Problems!

- Lack of standardized description format
  - Great blog posts, log signatures as unstructured text
  - No generic format like YARA or Snort rules
- Heterogeneous environments:
  - The *n+1 SIEMs* problem
  - Efficient distribution of log signatures for different systems
- Different SIEM products cover different signatures
- Vendor lock-in

SIGMA

- Generic signature format to describe interesting log events
- Open repository for Sigma signatures
- Converter that builds queries from Sigma signatures



Sigma Format

Generic Signature Description

Sigma Converter

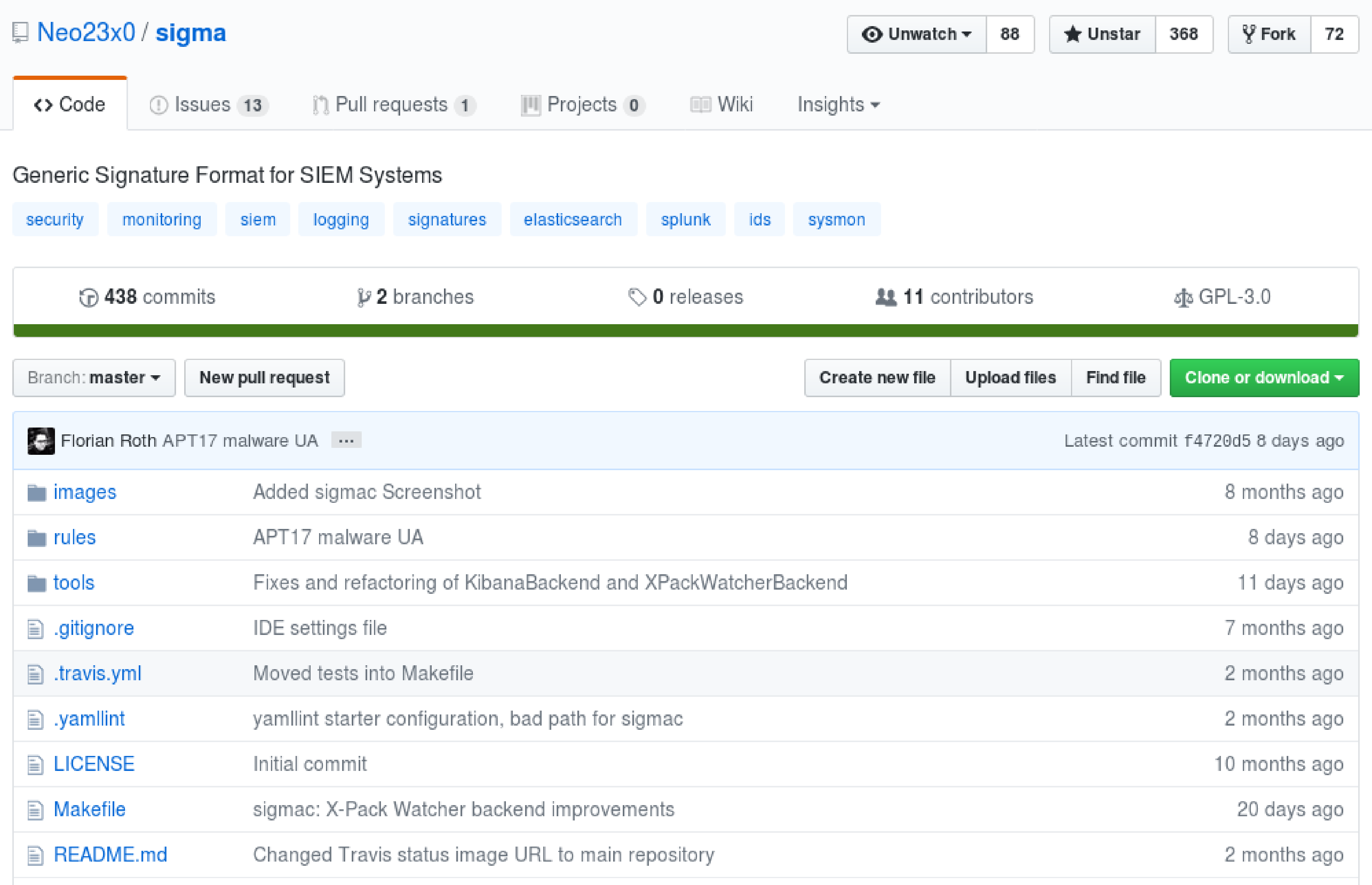
Applies Predefined and Custom Field Mapping

Elastic Search Queries

Splunk Searches

...

It’s open source!



Generic Signature Format for SIEM Systems

security monitoring siem logging signatures elasticsearch splunk ids sysmon

438 commits · 2 branches · 0 releases · 11 contributors · GPL-3.0

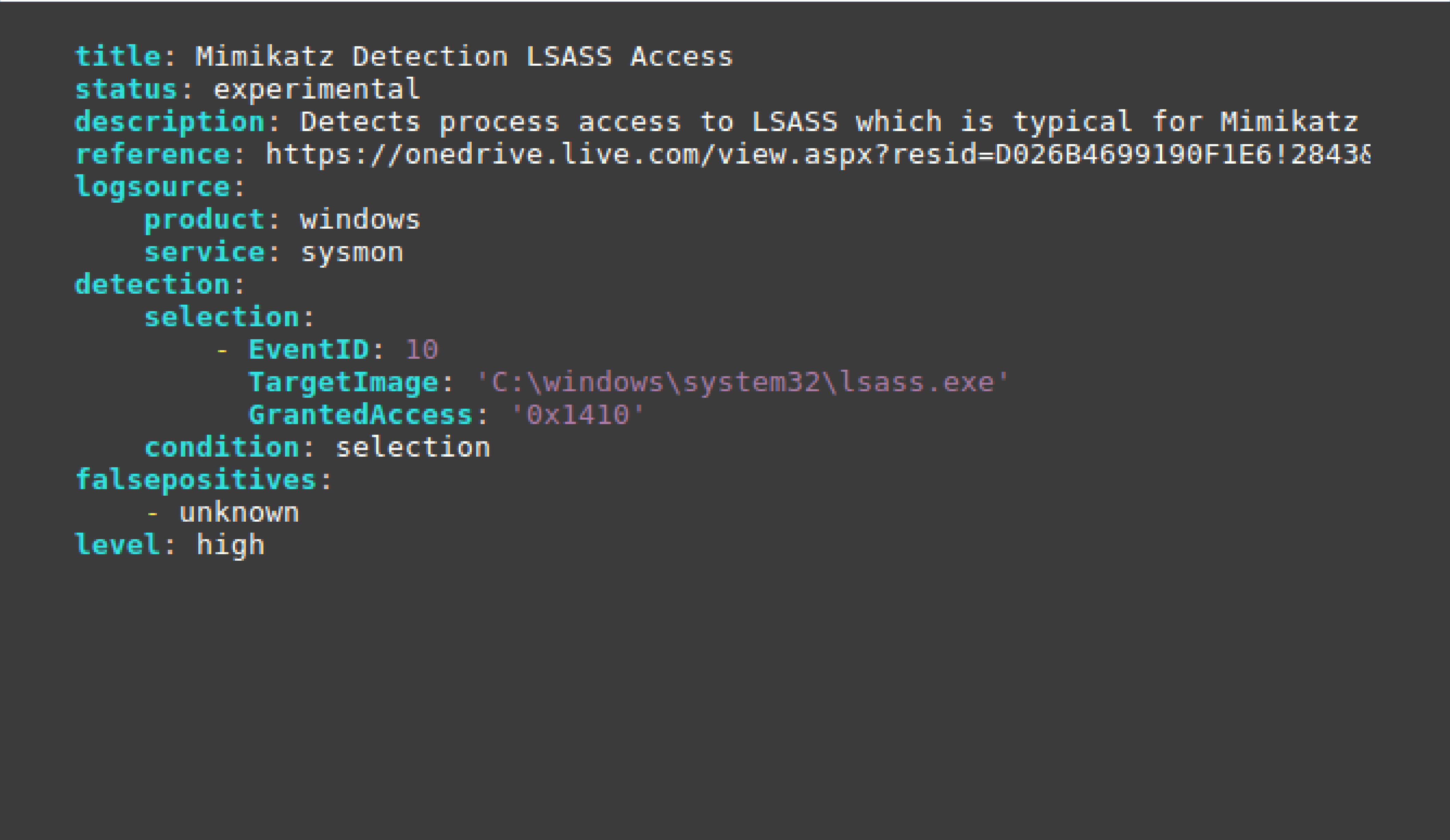
Florian Roth APT17 malware UA ··· Latest commit f4720d5 8 days ago

images	Added sigmac Screenshot	8 months ago
rules	APT17 malware UA	8 days ago
tools	Fixes and refactoring of KibanaBackend and XPackWatcherBackend	11 days ago
.gitignore	IDE settings file	7 months ago
.travis.yml	Moved tests into Makefile	2 months ago
.yamllint	yamllint starter configuration, bad path for sigmac	2 months ago
LICENSE	Initial commit	10 months ago
Makefile	sigmac: X-Pack Watcher backend improvements	20 days ago
README.md	Changed Travis status image URL to main repository	2 months ago

Rule Format

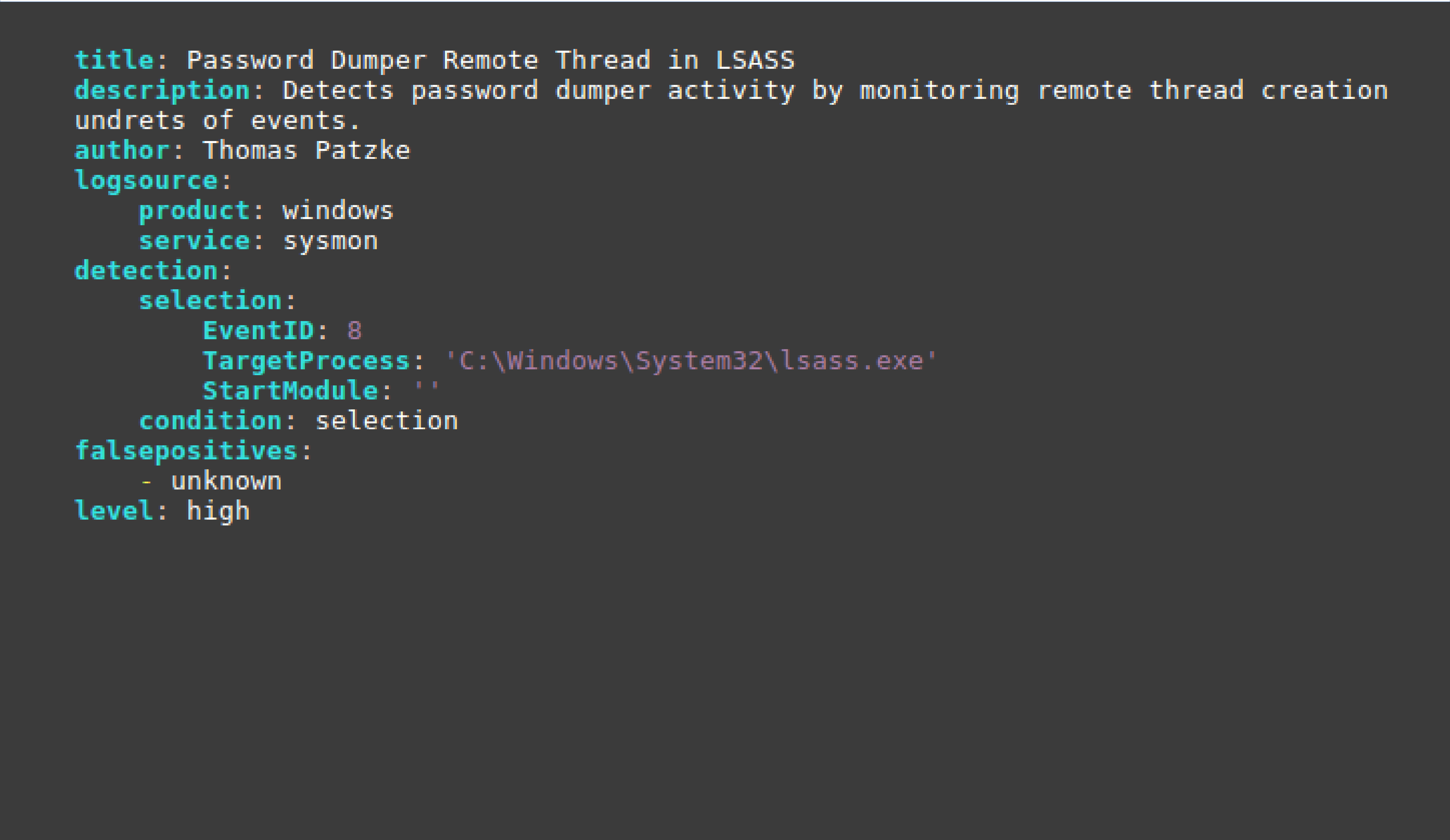
- Sigma rules are written in YAML
- Scope definition: which log sources are relevant?
- Search identifiers: Event IDs, values, strings
  - Lists of values
  - Key-value pairs that associate a log field with a value
- Condition:
  - Logical connection of search identifiers
  - Aggregation/correlation of matched events
- Metadata: title, description, author, state, (severity) level, reference, hints for identification of false positives

Rule Example: Mimikatz Detection



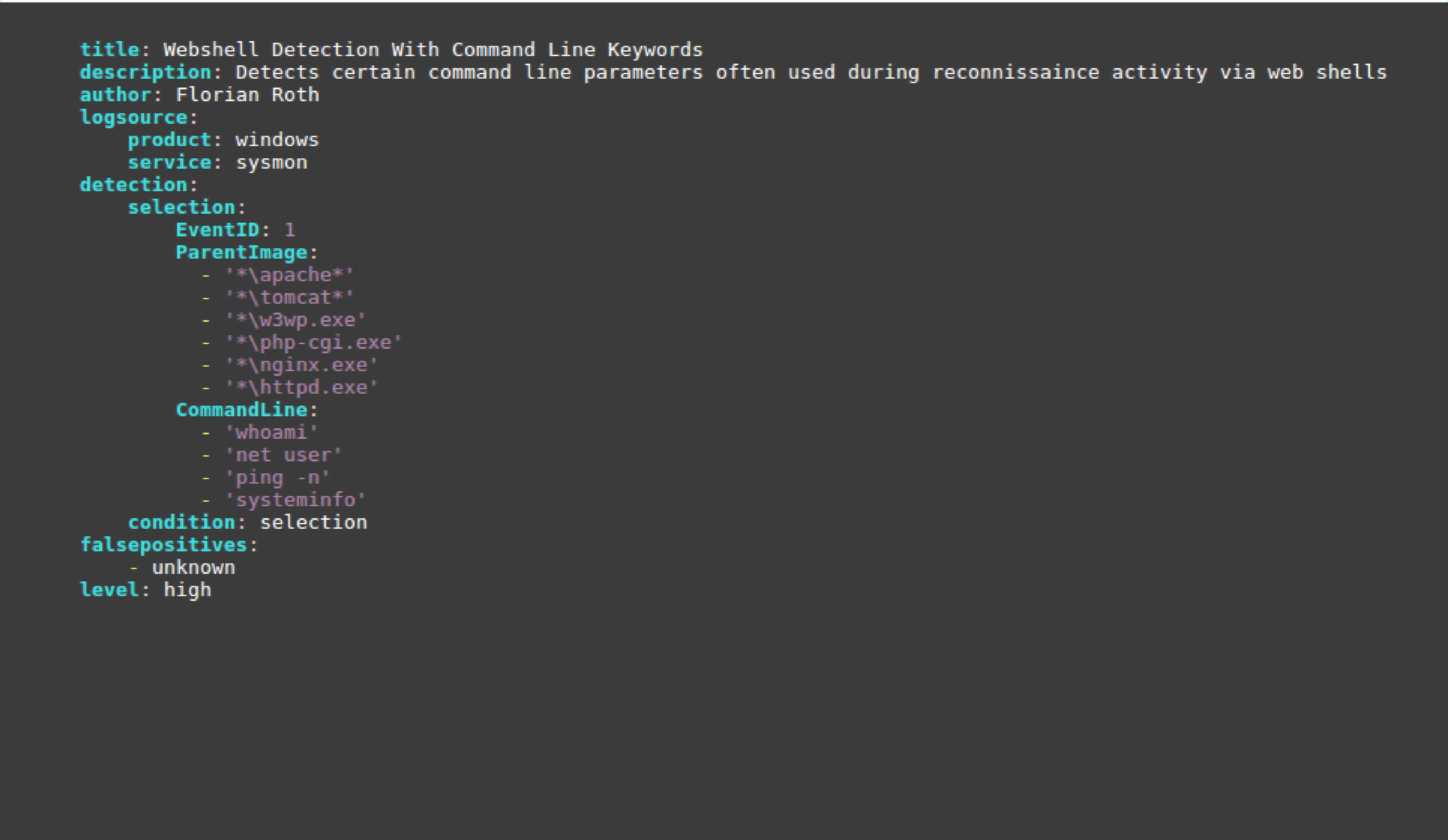
```
title: Mimikatz Detection LSASS Access
status: experimental
description: Detects process access to LSASS which is typical for Mimikatz
reference: https://onedrive.live.com/view.aspx?resid=D026B4699190F1E6!28438
logsource:
    product: windows
    service: sysmon
detection:
    selection:
        - EventID: 10
          TargetImage: 'C:\windows\system32\lsass.exe'
          GrantedAccess: '0x1410'
    condition: selection
falsepositives:
    - unknown
level: high
```

Rule Example:
WCE Detection



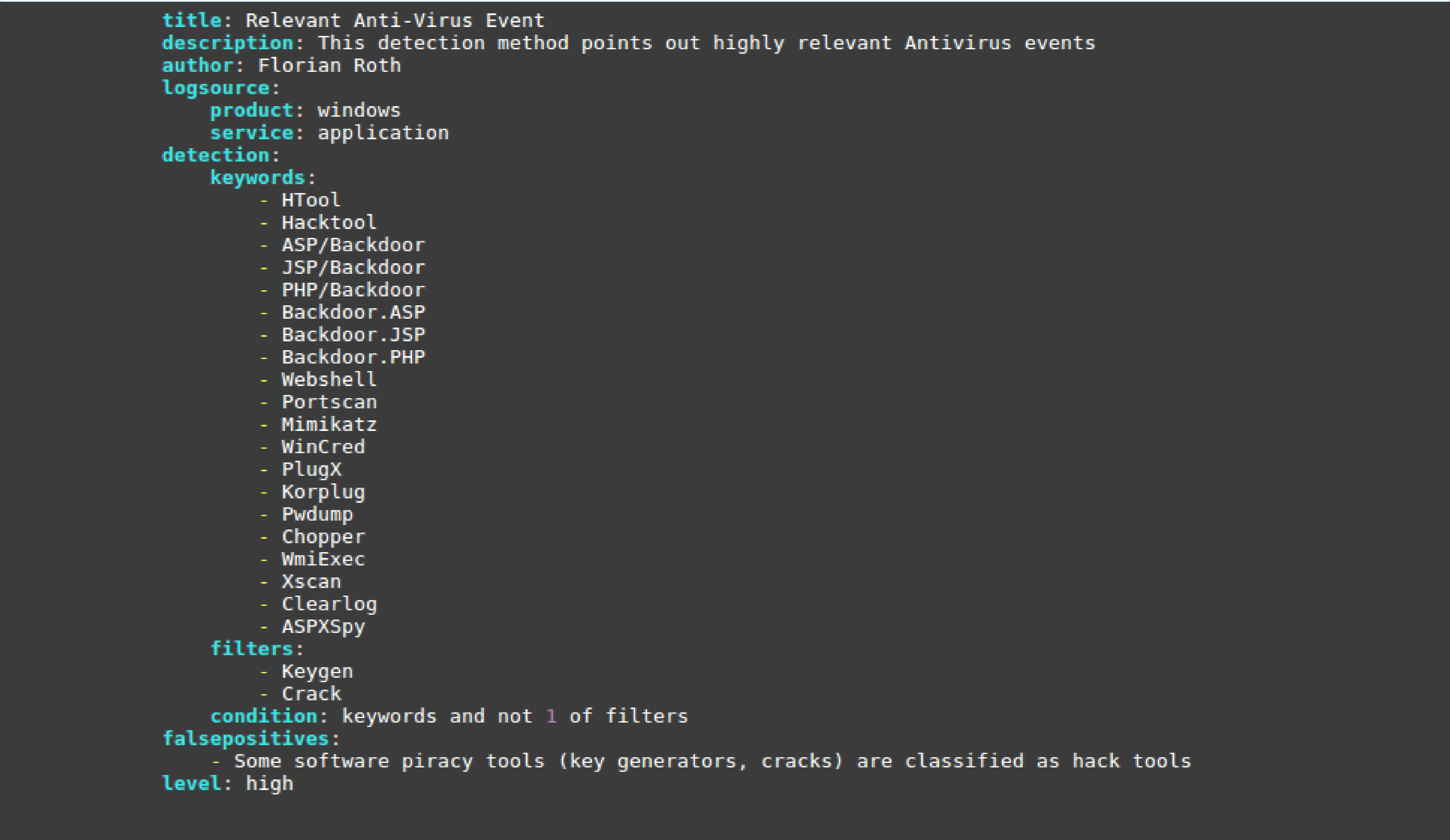
```
title: Password Dumper Remote Thread in LSASS
description: Detects password dumper activity by monitoring remote thread creation
undrets of events.
author: Thomas Patzke
logsource:
    product: windows
    service: sysmon
detection:
    selection:
        EventID: 8
        TargetProcess: 'C:\Windows\System32\lsass.exe'
        StartModule: ''
    condition: selection
falsepositives:
    - unknown
level: high
```

Rule Example: Webshell Reconnaissance Activity



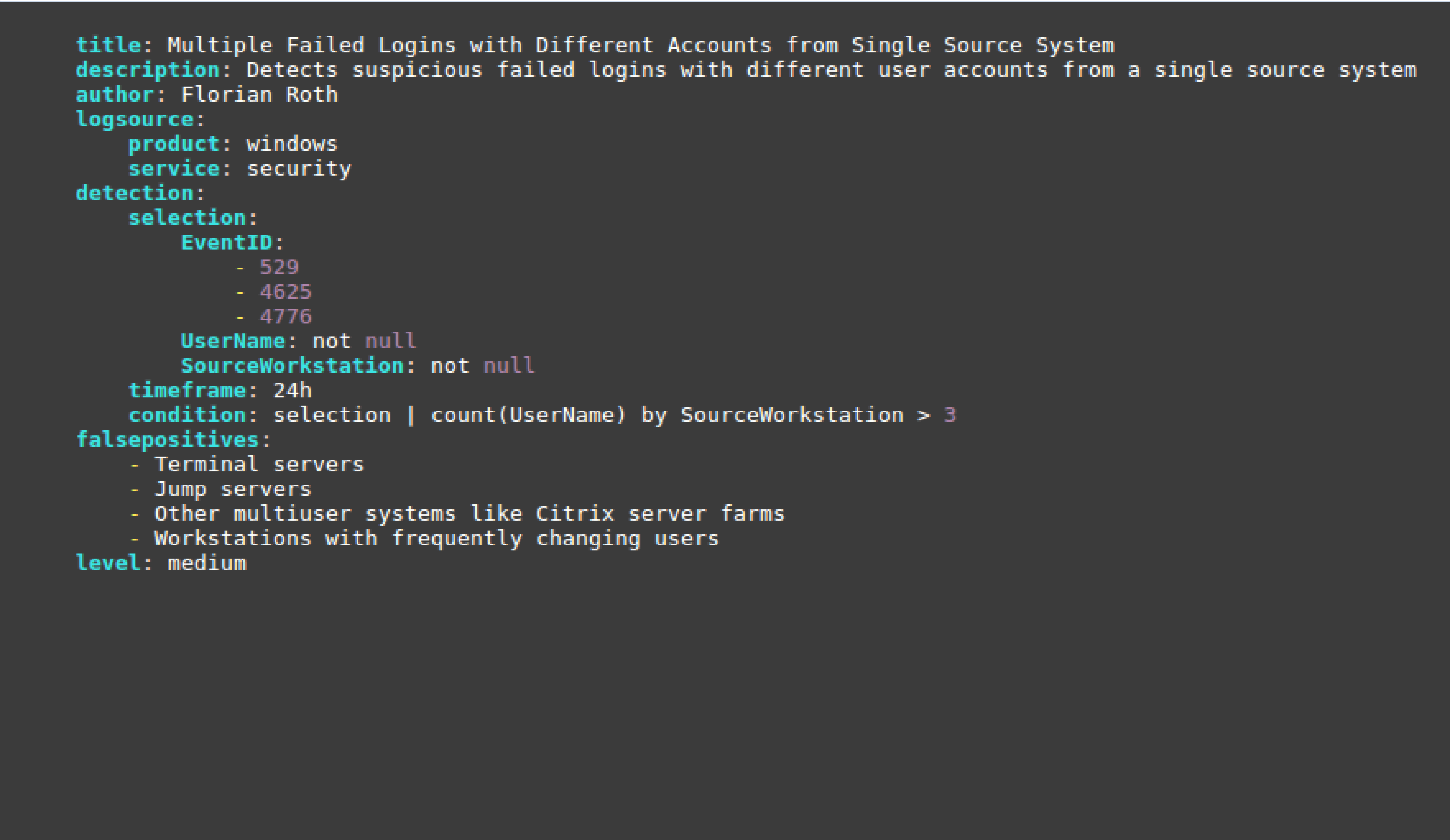
```
title: Webshell Detection With Command Line Keywords
description: Detects certain command line parameters often used during reconnissaince activity via web shells
author: Florian Roth
logsource:
    product: windows
    service: sysmon
detection:
    selection:
        EventID: 1
        ParentImage:
          - '*\apache*'
          - '*\tomcat*'
          - '*\w3wp.exe'
          - '*\php-cgi.exe'
          - '*\nginx.exe'
          - '*\httpd.exe'
        CommandLine:
          - 'whoami'
          - 'net user'
          - 'ping -n'
          - 'systeminfo'
    condition: selection
falsepositives:
    - unknown
level: high
```

Rule Example: Relevant AV Events



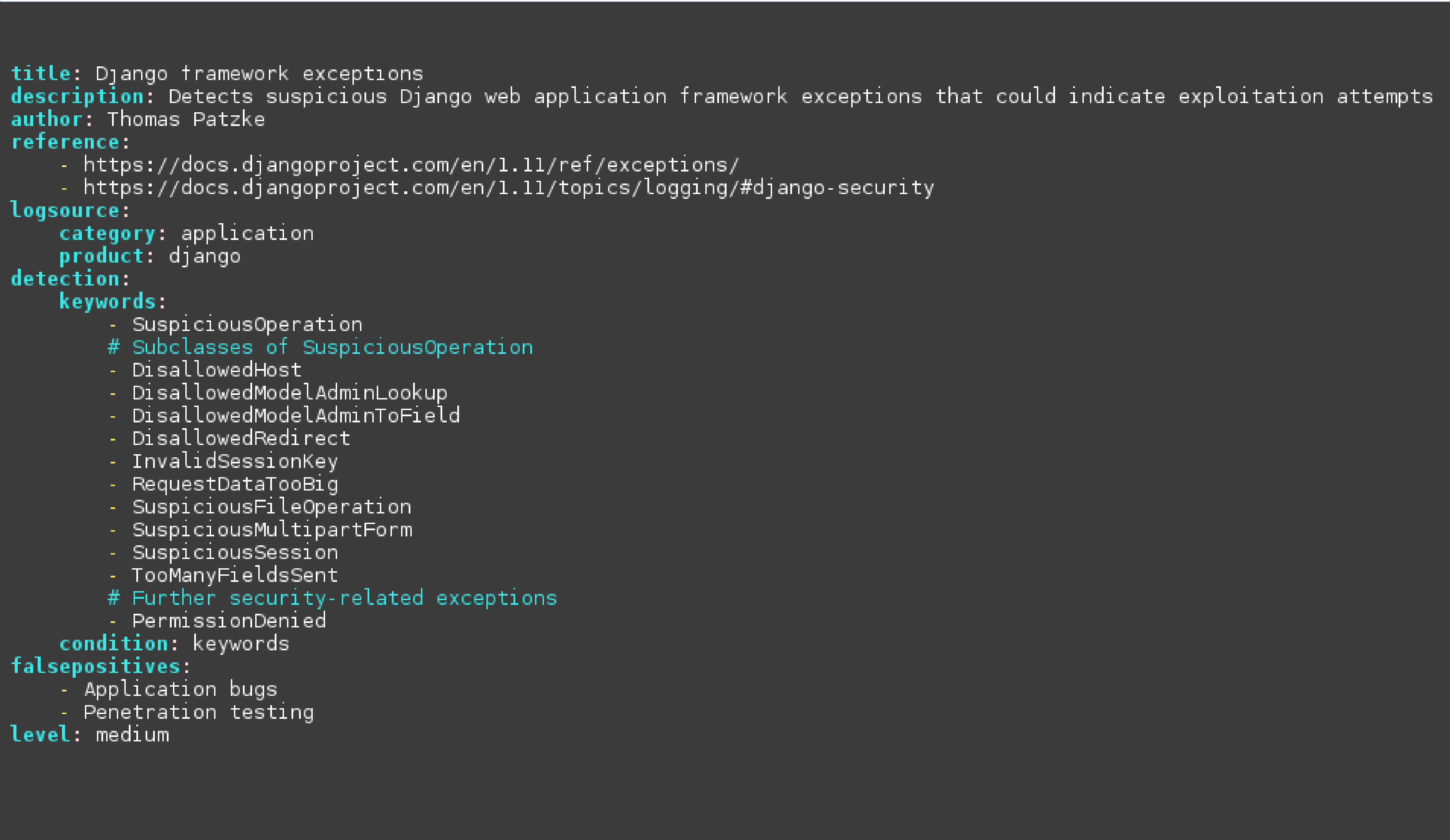
```
title: Relevant Anti-Virus Event
description: This detection method points out highly relevant Antivirus events
author: Florian Roth
logsource:
    product: windows
    service: application
detection:
    keywords:
        - HTool
        - Hacktool
        - ASP/Backdoor
        - JSP/Backdoor
        - PHP/Backdoor
        - Backdoor.ASP
        - Backdoor.JSP
        - Backdoor.PHP
        - Webshell
        - Portscan
        - Mimikatz
        - WinCred
        - PlugX
        - Korplug
        - Pwdump
        - Chopper
        - WmiExec
        - Xscan
        - Clearlog
        - ASPXSpy
    filters:
        - Keygen
        - Crack
    condition: keywords and not 1 of filters
falsepositives:
    - Some software piracy tools (key generators, cracks) are classified as hack tools
level: high
```

Rule Example:
Suspicious Login Attempts



```
title: Multiple Failed Logins with Different Accounts from Single Source System
description: Detects suspicious failed logins with different user accounts from a single source system
author: Florian Roth
logsource:
    product: windows
    service: security
detection:
    selection:
        EventID:
            - 529
            - 4625
            - 4776
        UserName: not null
        SourceWorkstation: not null
    timeframe: 24h
    condition: selection | count(UserName) by SourceWorkstation > 3
falsepositives:
    - Terminal servers
    - Jump servers
    - Other multiuser systems like Citrix server farms
    - Workstations with frequently changing users
level: medium
```

Example: Django Exceptions



```
title: Django framework exceptions
description: Detects suspicious Django web application framework exceptions that could indicate exploitation attempts
author: Thomas Patzke
reference:
    - https://docs.djangoproject.com/en/1.11/ref/exceptions/
    - https://docs.djangoproject.com/en/1.11/topics/logging/#django-security
logsource:
    category: application
    product: django
detection:
    keywords:
        - SuspiciousOperation
        # Subclasses of SuspiciousOperation
        - DisallowedHost
        - DisallowedModelAdminLookup
        - DisallowedModelAdminToField
        - DisallowedRedirect
        - InvalidSessionKey
        - RequestDataTooBig
        - SuspiciousFileOperation
        - SuspiciousMultipartForm
        - SuspiciousSession
        - TooManyFieldsSent
        # Further security-related exceptions
        - PermissionDenied
    condition: keywords
falsepositives:
    - Application bugs
    - Penetration testing
level: medium
```

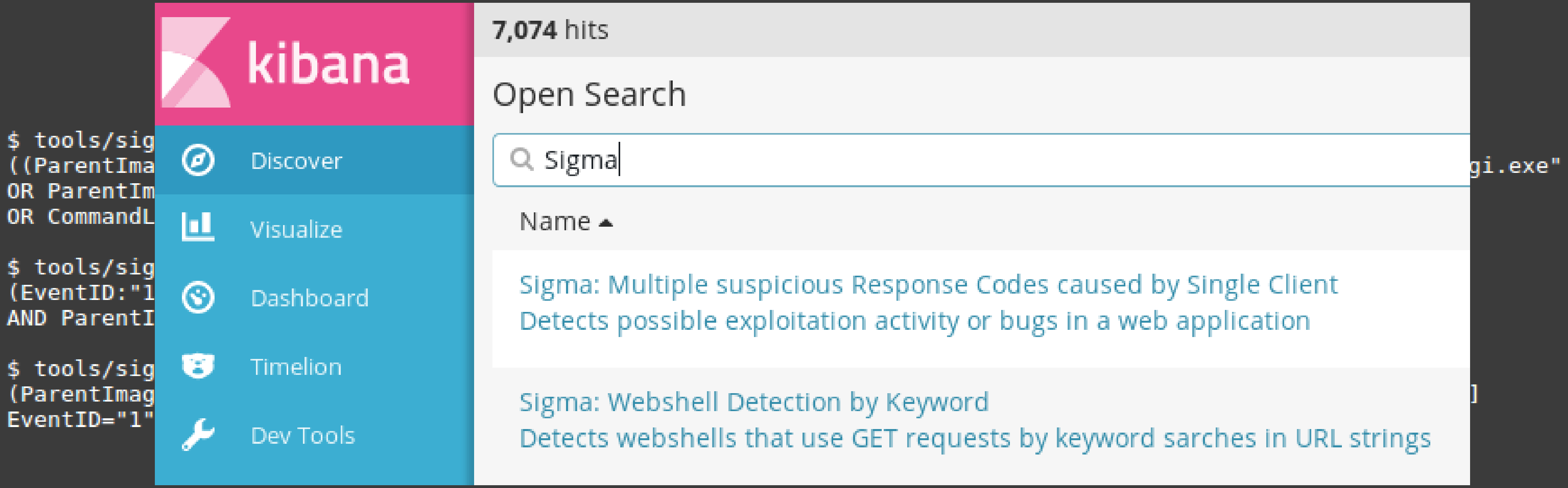
Sigma Converter

Conversion of a Sigma rule into three different query languages:

- Splunk
- Elasticsearch
- LogPoint

Conversion to frontend/tool configurations:

- Kibana searches
- Elastic X-Pack Watcher alerts



```
$ tools/sig
((ParentIma                                                       gi.exe"
OR ParentIm
OR CommandL

$ tools/sig
(EventID:"1
AND ParentI

$ tools/sig
(ParentImag                                                       ]
EventID="1"
```

7,074 hits

Open Search

Sigma

Name ▲

Sigma: Multiple suspicious Response Codes caused by Single Client
Detects possible exploitation activity or bugs in a web application

Sigma: Webshell Detection by Keyword
Detects webshells that use GET requests by keyword sarches in URL strings

Challenges in Rule Conversion

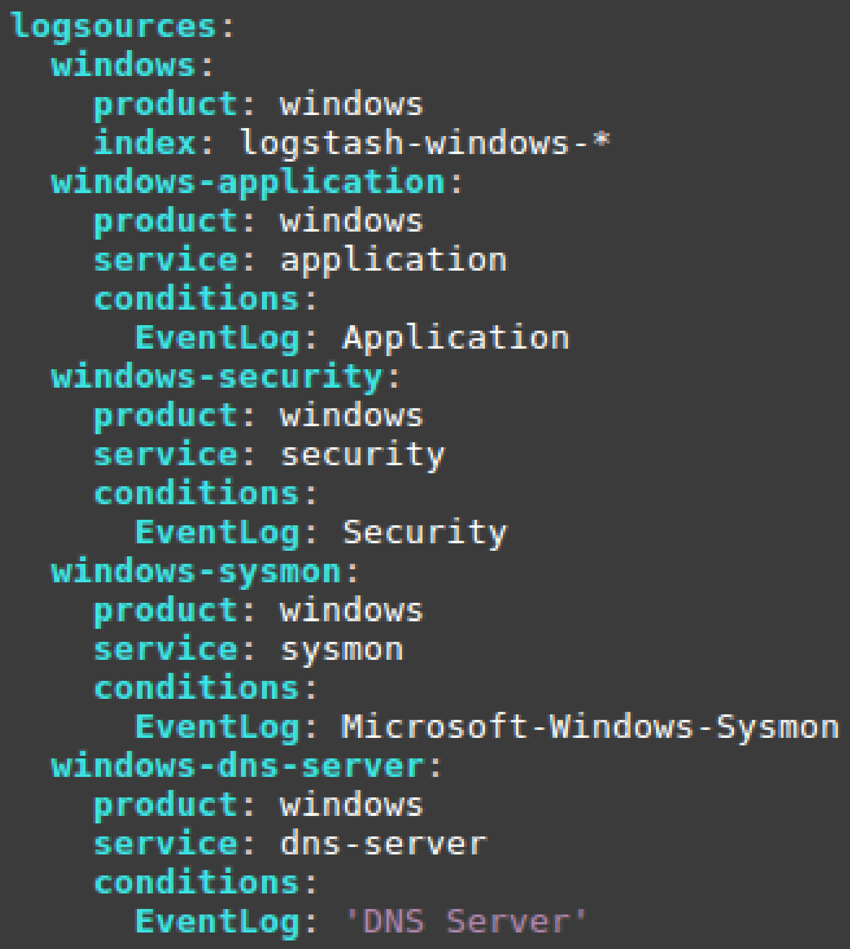
- Usage of different field names
  - Solution: field name mappings from Sigma rule field names to SIEM/environment specific names
- Inconsistent field names, multiple fields for one purpose
  - Solution: 1:n field name mappings
- Field names depend on event type, e.g. LogPoint has four names for *SubjectAccountName* or *UserName*.
  - Solution: Conditional field name mappings
- Log sources match to subsets of indexed log data: you don’t want to search web server logs for Windows security events
  - Solution: match category/product/service tuples to index patterns and conditions
- Rules refer to subsets of values which are environment-specific, e.g. client systems
  - Solution: place holders

Sigma Converter Configurations

- Sigma repository contains SIEM-specific configurations as start points
  - ELK
  - Splunk
  - Logpoint
- Environment-specific configuration must be added
- Sigma converter generates queries
  - with mapped field names
  - with additional conditions to narrow query to relevant data set

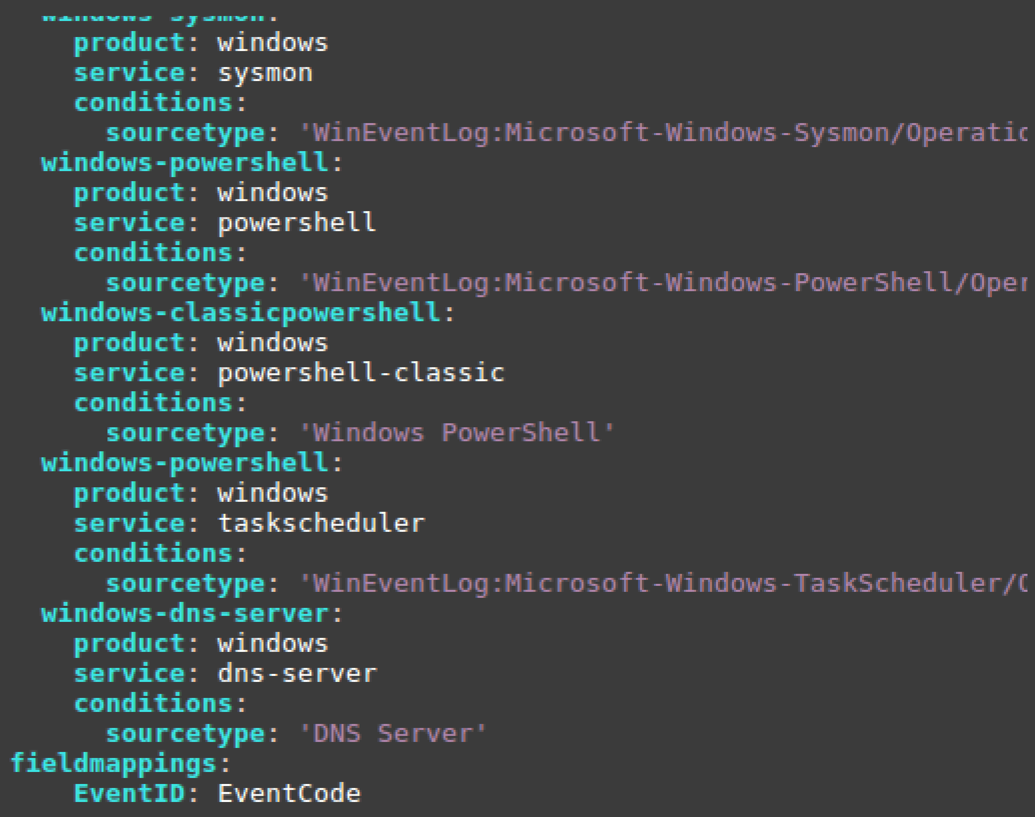
Sigma Converter Configurations

ELK



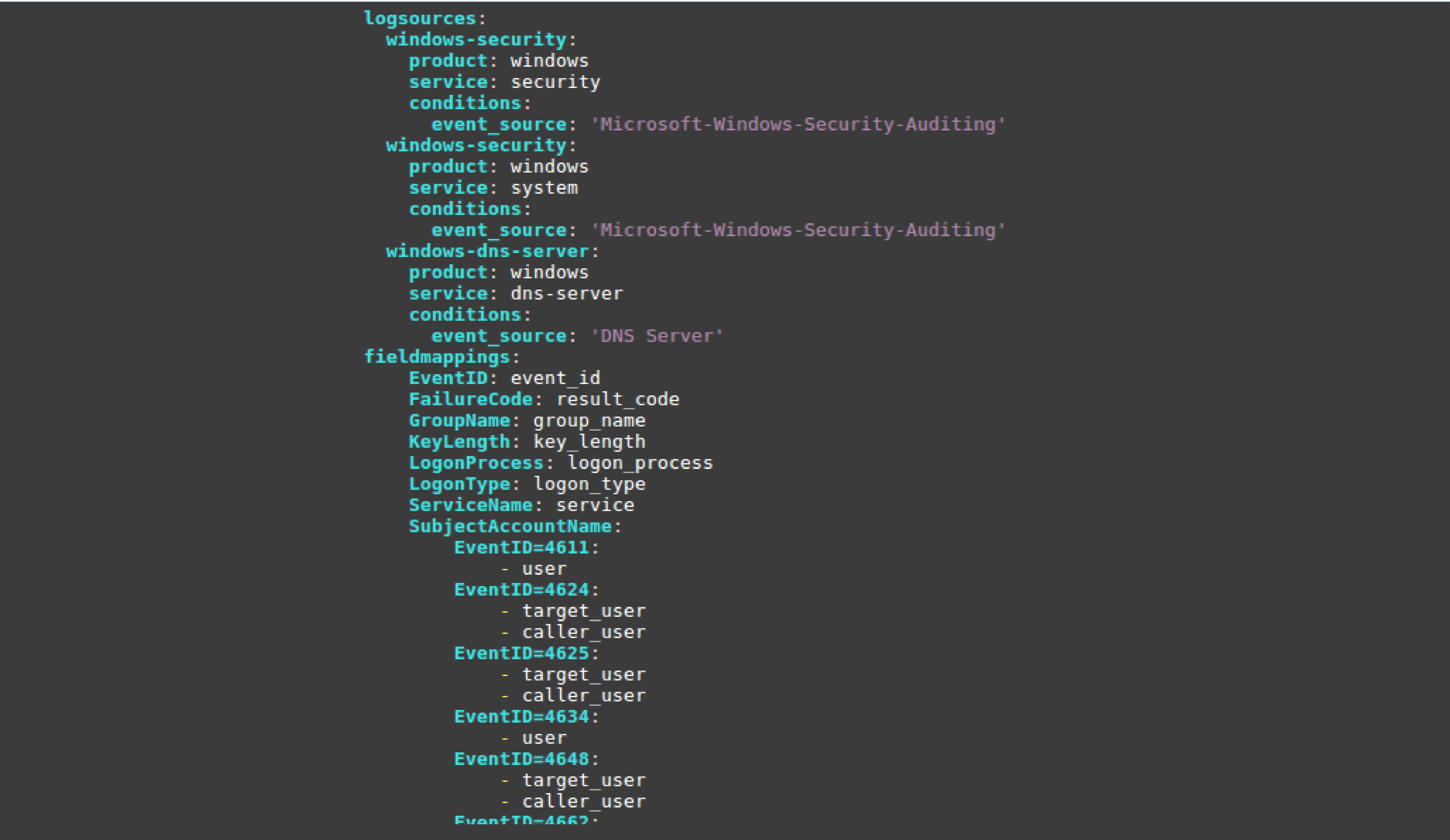
```
logsources:
  windows:
    product: windows
    index: logstash-windows-*
  windows-application:
    product: windows
    service: application
    conditions:
      EventLog: Application
  windows-security:
    product: windows
    service: security
    conditions:
      EventLog: Security
  windows-sysmon:
    product: windows
    service: sysmon
    conditions:
      EventLog: Microsoft-Windows-Sysmon
  windows-dns-server:
    product: windows
    service: dns-server
    conditions:
      EventLog: 'DNS Server'
```

Splunk



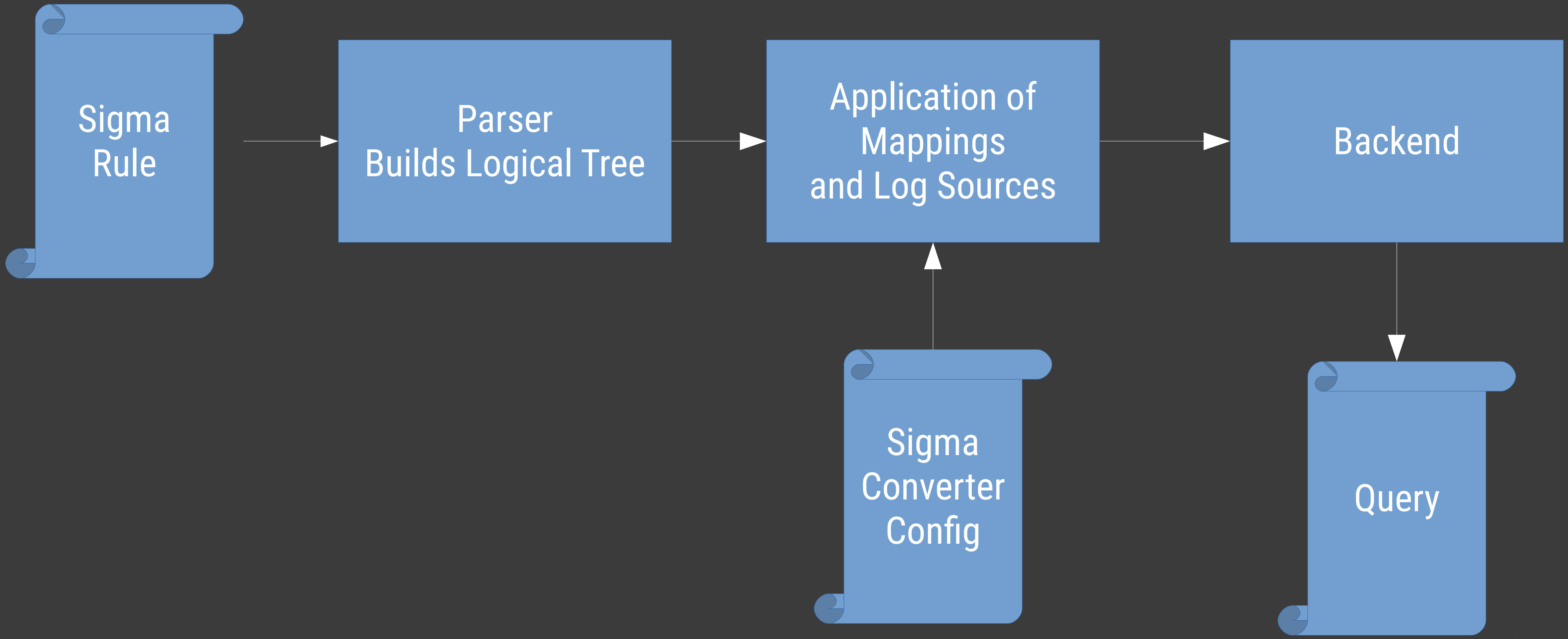
```
  windows-sysmon:
    product: windows
    service: sysmon
    conditions:
      sourcetype: 'WinEventLog:Microsoft-Windows-Sysmon/Operatio
  windows-powershell:
    product: windows
    service: powershell
    conditions:
      sourcetype: 'WinEventLog:Microsoft-Windows-PowerShell/Oper
  windows-classicpowershell:
    product: windows
    service: powershell-classic
    conditions:
      sourcetype: 'Windows PowerShell'
  windows-powershell:
    product: windows
    service: taskscheduler
    conditions:
      sourcetype: 'WinEventLog:Microsoft-Windows-TaskScheduler/O
  windows-dns-server:
    product: windows
    service: dns-server
    conditions:
      sourcetype: 'DNS Server'
fieldmappings:
    EventID: EventCode
```

Sigma Converter Logpoint Configuration



```
logsources:
  windows-security:
    product: windows
    service: security
    conditions:
      event_source: 'Microsoft-Windows-Security-Auditing'
  windows-security:
    product: windows
    service: system
    conditions:
      event_source: 'Microsoft-Windows-Security-Auditing'
  windows-dns-server:
    product: windows
    service: dns-server
    conditions:
      event_source: 'DNS Server'
fieldmappings:
    EventID: event_id
    FailureCode: result_code
    GroupName: group_name
    KeyLength: key_length
    LogonProcess: logon_process
    LogonType: logon_type
    ServiceName: service
    SubjectAccountName:
        EventID=4611:
            - user
        EventID=4624:
            - target_user
            - caller_user
        EventID=4625:
            - target_user
            - caller_user
        EventID=4634:
            - user
        EventID=4648:
            - target_user
            - caller_user
        EventID=4662:
```

Conversion Process



Sigma Rule

Parser
Builds Logical Tree

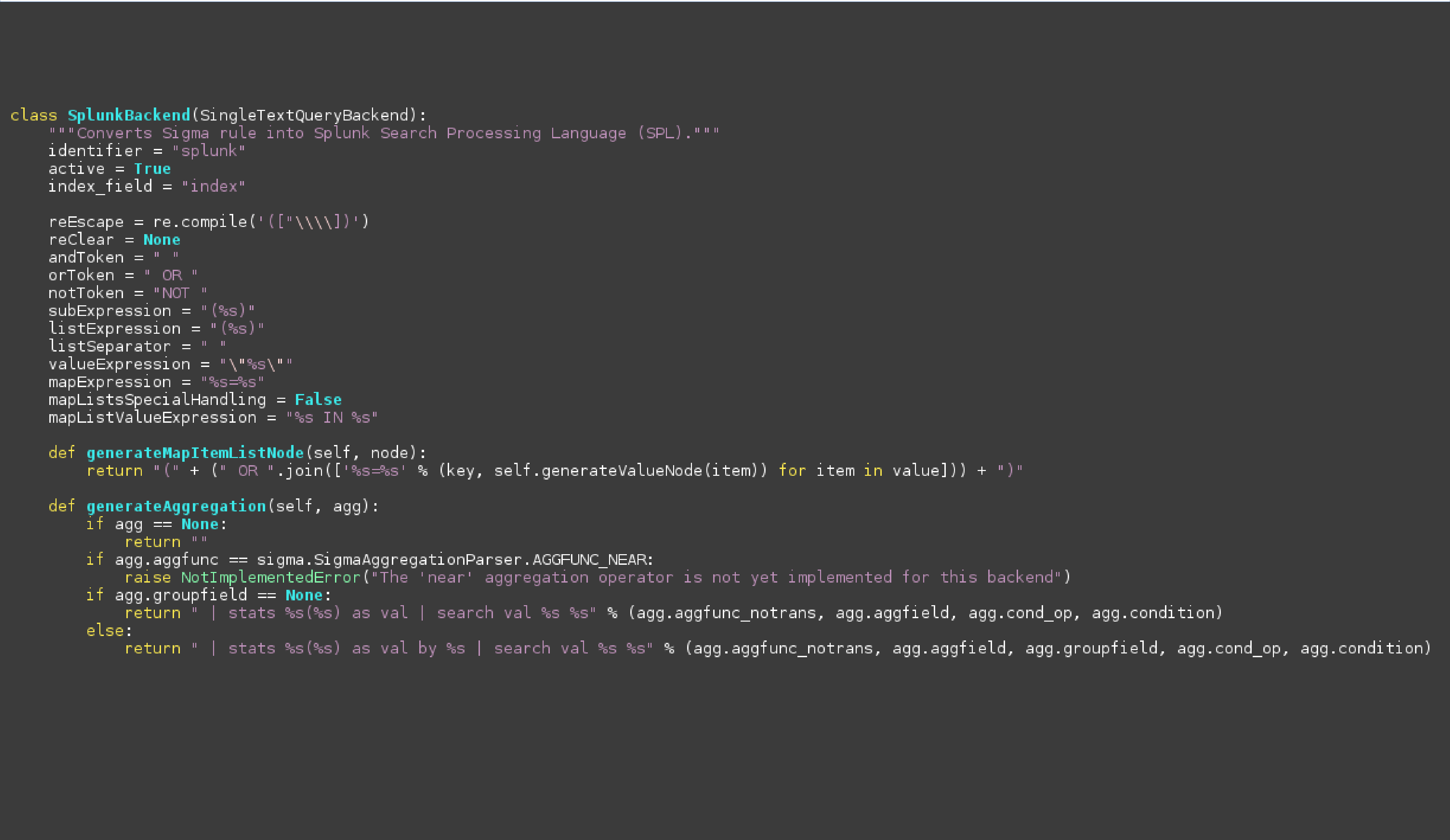
Application of Mappings and Log Sources

Backend

Sigma Converter Config

Query

Backend Implementation: Splunk



```
class SplunkBackend(SingleTextQueryBackend):
    """Converts Sigma rule into Splunk Search Processing Language (SPL)."""
    identifier = "splunk"
    active = True
    index_field = "index"

    reEscape = re.compile('(["\\\\])')
    reClear = None
    andToken = " "
    orToken = " OR "
    notToken = "NOT "
    subExpression = "(%s)"
    listExpression = "(%s)"
    listSeparator = " "
    valueExpression = "\"%s\""
    mapExpression = "%s=%s"
    mapListsSpecialHandling = False
    mapListValueExpression = "%s IN %s"

    def generateMapItemListNode(self, node):
        return "(" + (" OR ".join(['%s=%s' % (key, self.generateValueNode(item)) for item in value])) + ")"

    def generateAggregation(self, agg):
        if agg == None:
            return ""
        if agg.aggfunc == sigma.SigmaAggregationParser.AGGFUNC_NEAR:
            raise NotImplementedError("The 'near' aggregation operator is not yet implemented for this backend")
        if agg.groupfield == None:
            return " | stats %s(%s) as val | search val %s %s" % (agg.aggfunc_notrans, agg.aggfield, agg.cond_op, agg.condition)
        else:
            return " | stats %s(%s) as val by %s | search val %s %s" % (agg.aggfunc_notrans, agg.aggfield, agg.groupfield, agg.cond_op, agg.condition)
```

Contributors and Community

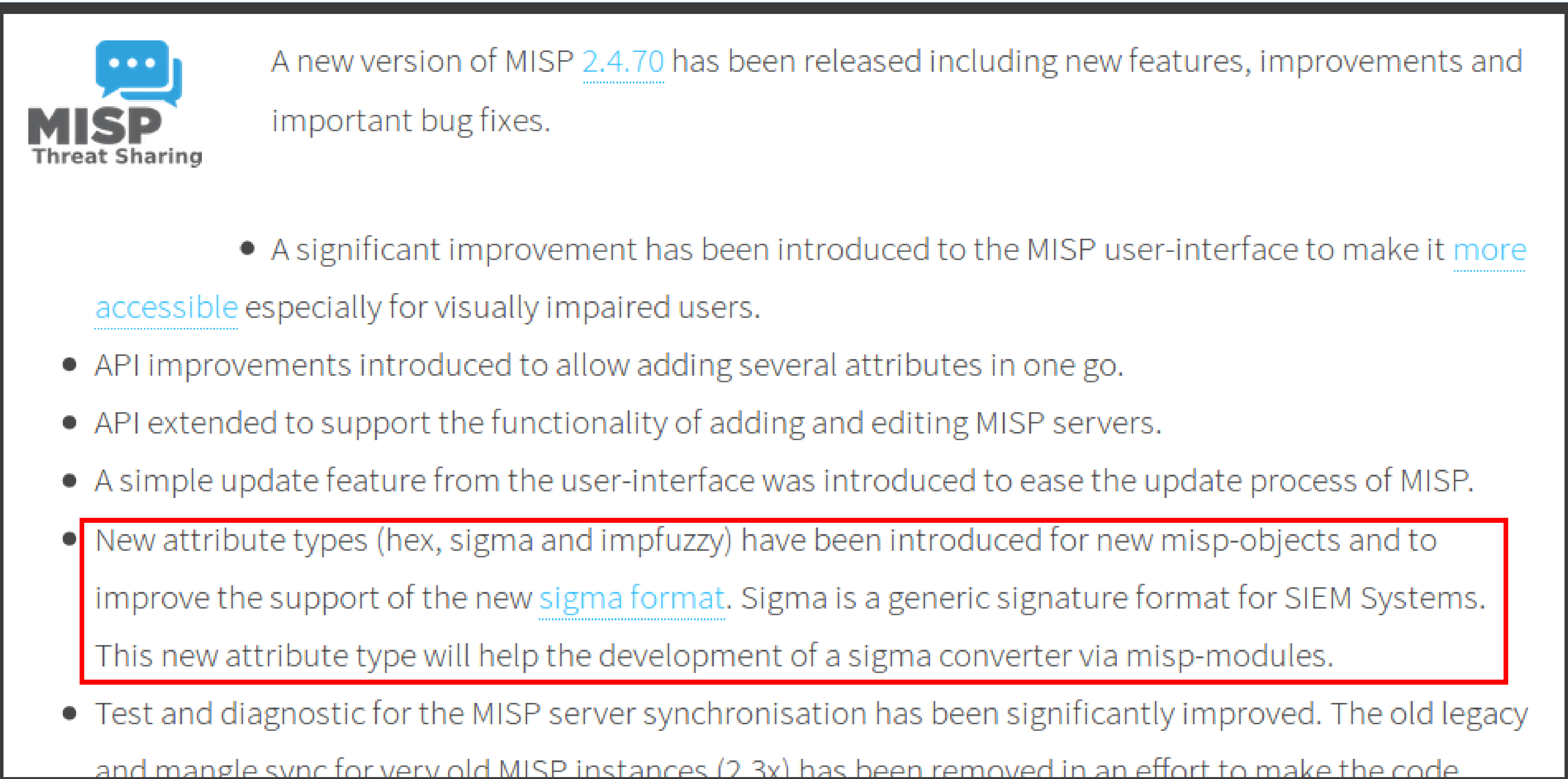
Github:

- Florian Roth: initiator, specification, main rule contributor, creates Sigma rules from blog articles before you've finished to reading them ;-)
- Thomas Patzke: Sigma converter, a few rules
- Michael Haag, Nate Guagenti, Ilias el Matani, Omer Yampel, Dimitrios Slamaris, yugoslavskiy:, @secman-pl created or improved rules
- Ben de Haan: LogPoint backend
- Devin Ferguson: X-Pack Watcher backend
- Julien (@juju4): Integration tests

Collaboration: Slack Channel, Ops Trust Community

- Discussion of current and general threats
- Collaboration on new rules
- Invite only: trusted exchange of sensitive information

Current State and Future Work



A new version of MISP 2.4.70 has been released including new features, improvements and important bug fixes.

- A significant improvement has been introduced to the MISP user-interface to make it more accessible especially for visually impaired users.
- API improvements introduced to allow adding several attributes in one go.
- API extended to support the functionality of adding and editing MISP servers.
- A simple update feature from the user-interface was introduced to ease the update process of MISP.
- New attribute types (hex, sigma and impfuzzy) have been introduced for new misp-objects and to improve the support of the new sigma format. Sigma is a generic signature format for SIEM Systems. This new attribute type will help the development of a sigma converter via misp-modules.
- Test and diagnostic for the MISP server synchronisation has been significantly improved. The old legacy and mangle sync for very old MISP instances (2.3x) has been removed in an effort to make the code

- Testing!

Questions?

- Rules + Code: https://github.com/Neo23x0/sigma
- Documentation: https://github.com/Neo23x0/sigma/wiki
- Thomas Patzke

  🐦 @blubbfiction

  ✉ thomas@patzke.org
- Florian Roth

  🐦 @cyb3rops