



# Threat Hunting with Application Logs and Sigma

Thomas Patzke, 14. November 2017

# Agenda

- Introduction to Threat Detection with Log Analysis
- Log Traces of Application Attacks
- Motivation for a Log Signature Format
- Sigma – The Open Source Approach
  - Rule Format
  - Rule Examples
  - Conversion to SIEM queries
- How can developers, pentesters and security researches contribute?

# From Attack to Logfile

# From Attack to Logfile

Attack Attempts

# From Attack to Logfile

Attack Attempts



Application Errors

# From Attack to Logfile

Attack Attempts



Application Errors



Error Logs

# From Attack to Logfile

Attack Attempts



Application Errors



Error Logs

This is something we are able to detect!

# **Threat Hunting: From Log File to Threat Detection**



# **Threat Hunting: From Log File to Threat Detection**

Log Files

# Threat Hunting: From Log File to Threat Detection

Log Files



No standardized structure

# Threat Hunting: From Log File to Threat Detection

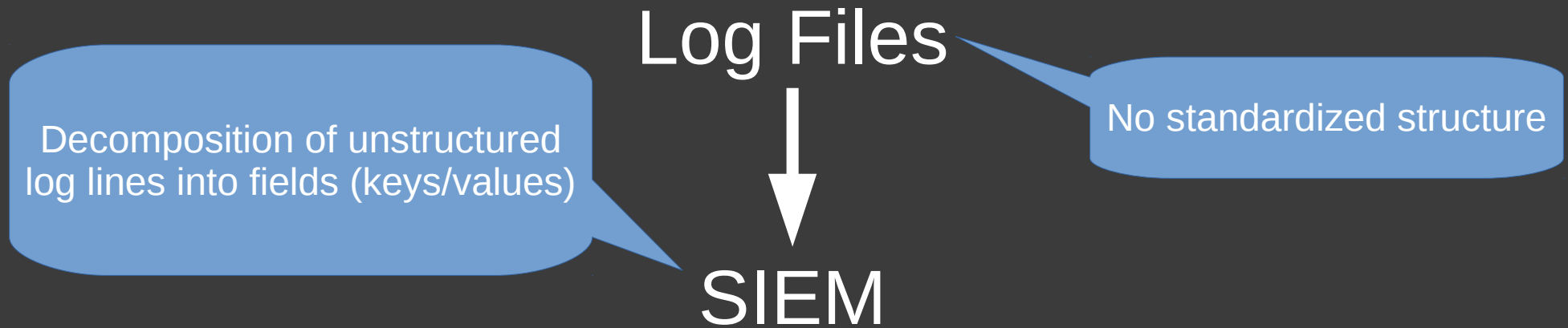
Log Files



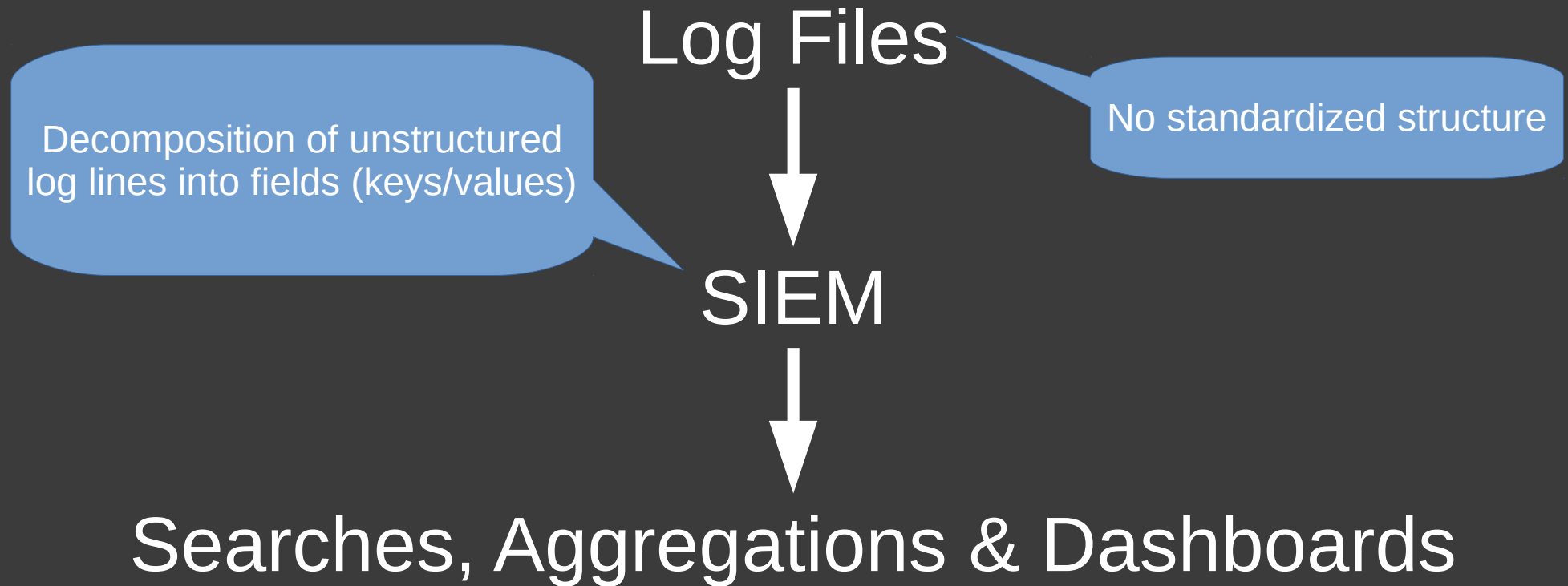
SIEM

No standardized structure

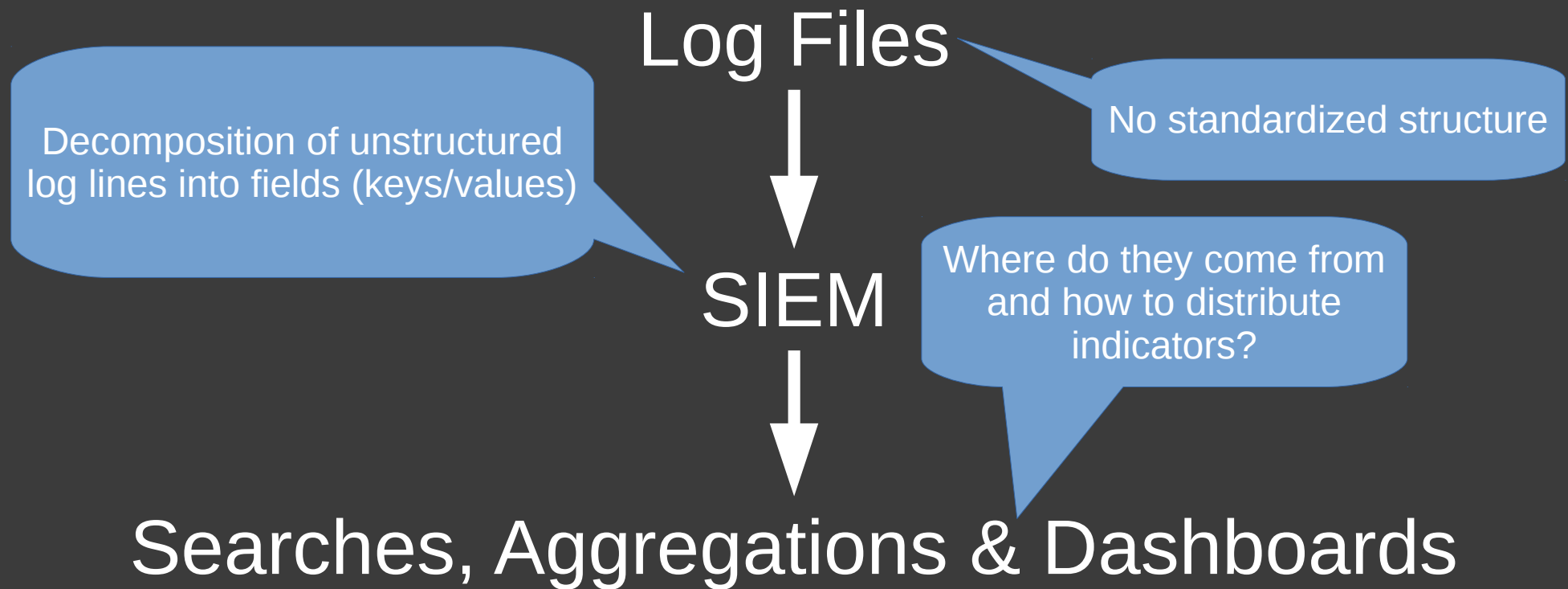
# Threat Hunting: From Log File to Threat Detection



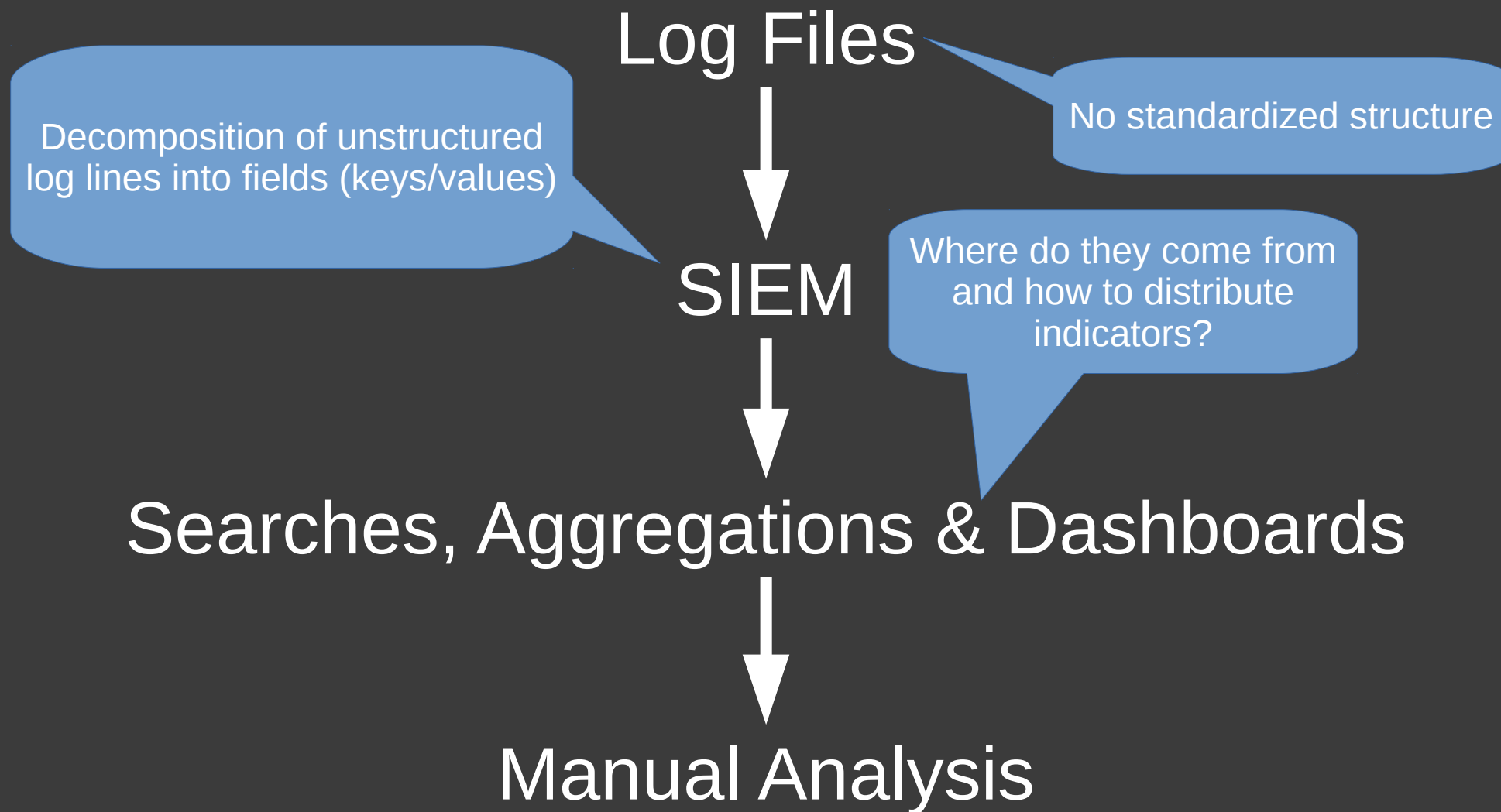
# Threat Hunting: From Log File to Threat Detection



# Threat Hunting: From Log File to Threat Detection



# Threat Hunting: From Log File to Threat Detection



# Threat Detection with Log Monitoring: Log Sources

- Firewall Logs
  - Successful/Filtered IP/TCP/UDP Communication
- Operating System Logs
  - Authentication
  - Process Execution
  - Resource Access
- Proxy Logs
- Web Server Access Logs



# **Threat Detection with Log Monitoring: Signature Examples**

# Threat Detection with Log Monitoring: Signature Examples

- Authentication & Accounts:
  - Large number of failed logon attempts
  - Alternation and usage of specific accounts (e.g. DSRM)
  - SID history

# Threat Detection with Log Monitoring: Signature Examples

- 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

# Threat Detection with Log Monitoring: Signature Examples

- 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

# Threat Detection with Log Monitoring: Signature Examples

- 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)

# **Threat Detection with Log Monitoring: Application Events**

# Threat Detection with Log Monitoring: Application Events

- Web Server Access Logs:
  - 4xx Errors: Enumeration and Reconnaissance activity
  - 5xx Errors: Exploitation

# Threat Detection with Log Monitoring: Application Events

- Web Server Access Logs:
  - 4xx Errors: Enumeration and Reconnaissance activity
  - 5xx Errors: Exploitation
- Application Error Logs
  - Exceptions
  - Specific messages



# **Some Application Error Examples**

# Some Application Error Examples

## OpenSSH

- “unexpected internal error”
- “error in libcrypto”
- “unexpected bytes remain after decoding”

# Some Application Error Examples

## OpenSSH

- “unexpected internal error”
- “error in libcrypto”
- “unexpected bytes remain after decoding”

## vsftpd

- “weird status”
- “Input line too long”
- “syscall validate failed”

# Some Application Error Examples

## OpenSSH

- “unexpected internal error”

## vsftpd

- “weird status”

### SuspiciousOperation

`exception SuspiciousOperation[source]`

The SuspiciousOperation exception is raised when a user has performed an operation that should be considered suspicious from a security perspective, such as tampering with a session cookie. Subclasses of **SuspiciousOperation** include:

- **DisallowedHost**
- **DisallowedModelAdminLookup**
- **DisallowedModelAdminToField**
- **DisallowedRedirect**
- **InvalidSessionKey**
- **RequestDataTooBig**
- **SuspiciousFileOperation**
- **SuspiciousMultipartForm**
- **SuspiciousSession**
- **TooManyFieldsSent**

# Some Application Error Examples

```
Query: 'a'
[remote ] [2017-11-03 21:37:02,485] ERROR in app: Exception on / [GET]
[remote ] Traceback (most recent call last):
[remote ]   File "/usr/local/lib/python3.5/dist-packages/flask/app.py", line 1982, in wsgi_app
[remote ]     response = self.full_dispatch_request()
[remote ]   File "/usr/local/lib/python3.5/dist-packages/flask/app.py", line 1614, in full_dispatch_request
[remote ]     rv = self.handle_user_exception(e)
[remote ]   File "/usr/local/lib/python3.5/dist-packages/flask/app.py", line 1517, in handle_user_exception
[remote ]     reraise(exc_type, exc_value, tb)
[remote ]   File "/usr/local/lib/python3.5/dist-packages/flask/_compat.py", line 33, in reraise
[remote ]     raise value
[remote ]   File "/usr/local/lib/python3.5/dist-packages/flask/app.py", line 1612, in full_dispatch_request
[remote ]     rv = self.dispatch_request()
[remote ]   File "/usr/local/lib/python3.5/dist-packages/flask/app.py", line 1598, in dispatch_request
[remote ]     return self.view_functions[rule.endpoint](**req.view_args)
[remote ]   File "/var/www/VulnerableWebApp/Vulnerable.py", line 17, in entrypoint
[remote ]     cur.execute("SELECT ProductName, UnitPrice, CategoryName, CategoryDescription, SupplierName, SU
[remote ]
[remote ] sqlite3.OperationalError: unrecognized token: ""
```

- **DisallowedModelAdminToField**
- **DisallowedRedirect**
- **InvalidSessionKey**
- **RequestDataTooBig**
- **SuspiciousFileOperation**
- **SuspiciousMultipartForm**
- **SuspiciousSession**
- **TooManyFieldsSent**

# Problems?

# Problems?

## Windows Event Monitoring Guidance

### Recommended Events to Collect

#### Account Usage

User account information can be collected and audited. Tracking local account usage can help detect Pass the Hash activity and other unauthorized account usage. Additional information such as remote desktop logins, users added to privileged groups, and account lockouts can also be tracked. User accounts being promoted to privileged groups should be audited very 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	Microsoft-Windows-Security-Auditing
Account Name Changed	4781	Information	Security	Microsoft-Windows-Security-Auditing
Account removed from Local Sec. Grp.	4733	Information	Security	Microsoft-Windows-Security-Auditing

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

# 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	Microsoft-Windows-Security-Auditing
Account Name Changed	4781	Information	Security	Microsoft-Windows-Security-Auditing
Account removed from Local Sec. Grp.	4733	Information	Security	Microsoft-Windows-Security-Auditing

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

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



# 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:

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://adsecurity.org/?p=1714>

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

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

# Problems!

# Problems!

- Lack of standardized description format
  - Great blog posts, log signatures as unstructured text
  - No generic format like YARA or Snort rules

# 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

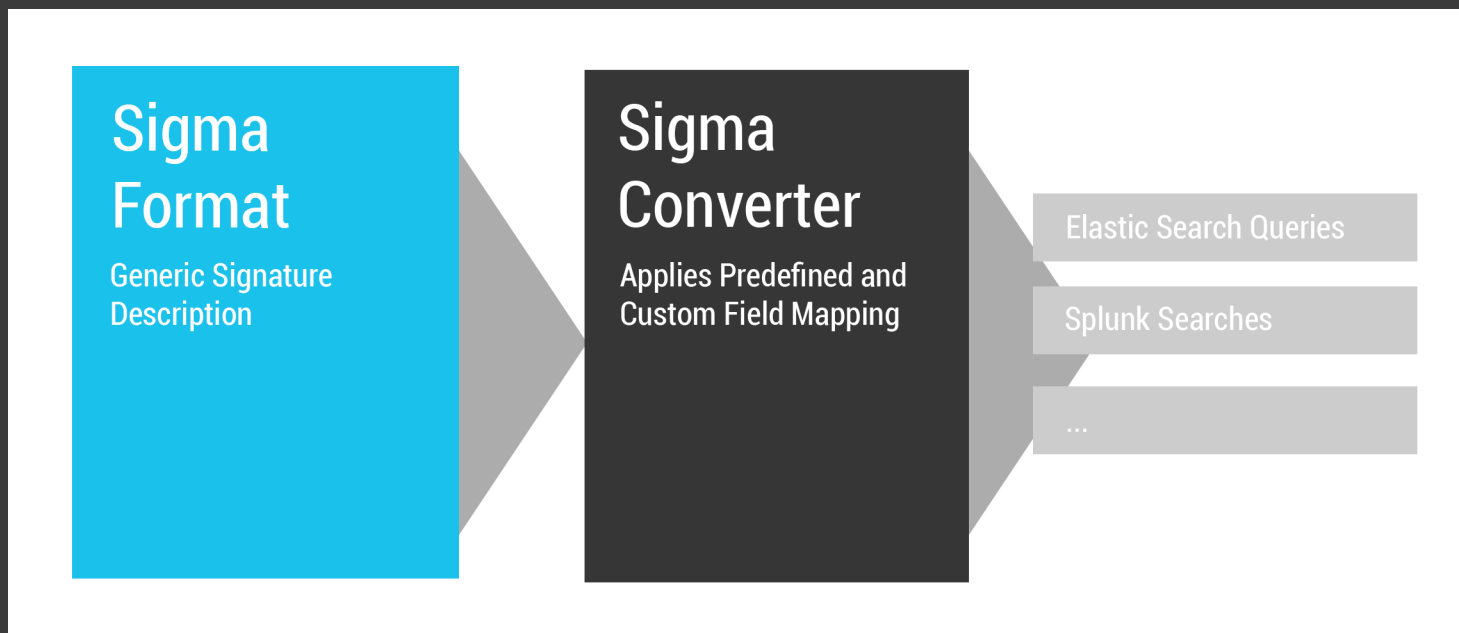
# Problems!

- 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

# 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

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



# It's open source!

Neo23x0 / **sigma** Unwatch 100 Unstar 446 Fork 85


[Code](#) [Issues 12](#) [Pull requests 1](#) [Projects 0](#) [Wiki](#) [Insights](#)

## Generic Signature Format for SIEM Systems

[security](#) [monitoring](#) [siem](#) [logging](#) [signatures](#) [elasticsearch](#) [splunk](#) [ids](#) [sysmon](#)

[517 commits](#) [3 branches](#) [0 releases](#) [11 contributors](#) [GPL-3.0](#)

Branch: **master** [New pull request](#) [Create new file](#) [Upload files](#) [Find file](#) [Clone or download](#)

 Florian Roth Bugfix in Adwind rule - typo in typo Latest commit 3a378f0 3 days ago

<a href="#">images</a>	Added sigmac Screenshot	9 months ago
<a href="#">rules</a>	Bugfix in Adwind rule - typo in typo	3 days ago
<a href="#">tests</a>	Improved testing	12 days ago
<a href="#">tools</a>	Added default index configs for usual ELK setups	5 days ago
<a href="#">.gitignore</a>	IDE settings file	8 months ago
<a href="#">.travis.yml</a>	Measurement of test coverage	26 days ago
<a href="#">.yamllint</a>	yamllint starter configuration, bad path for sigmac	4 months ago



# Rule Format

# Rule Format

- Sigma rules are written in YAML

# Rule Format

- Sigma rules are written in YAML
- Scope definition: which log sources are relevant?

# 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

# 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

# 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

# Example: HTTP Error Codes

```
title: Multiple suspicious Response Codes caused by Single Client
description: Detects possible exploitation activity or bugs in a web application
author: Thomas Patzke
logsource:
  category: webservers
detection:
  selection:
    response:
      - 400
      - 401
      - 403
      - 500
    timeframe: 10m
    condition: selection | count() by clientip > 10
fields:
  - client_ip
  - vhost
  - url
  - response
falsepositives:
  - Unstable application
  - Application that misuses the response codes
level: medium
```

# Example: HTTP Error Codes

**title:** Multiple suspicious Response Codes caused by Single Client  
**description:** Detects possible exploitation activity or bugs in a web application  
**author:** Thomas Patzke  
**logsource:**  
    **category:** webserver  
**detection:**  
    **selection:**  
        **response:**  
            - 400  
            - 401  
            - 403  
            - 500  
    **timeframe:** 10m  
    **condition:** selection | count() by clientip > 10

```
=== splunk ===  
((response="400" OR response="401" OR response="403" OR response="500")) | stats count() as val by clientip | search val > 10  
=== logpoint ===  
(response IN ["400", "401", "403", "500"]) | chart count() as val by clientip | search val > 10  
=== grep ===  
grep -P '^(?:.*(?:.*400|.*401|.*403|.*500))'
```

**falsepositives:**  
    - Unstable application  
    - Application that misuses the response codes  
**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
```

# Example: Django Exceptions

```
=== es-qs ===
("SuspiciousOperation" OR "DisallowedHost" OR "DisallowedModelAdminLookup" OR "Disa
llowedModelAdminToField" OR "DisallowedRedirect" OR "InvalidSessionKey" OR "Request
DataTooBig" OR "SuspiciousFileOperation" OR "SuspiciousMultipartForm" OR "Suspiciou
sSession" OR "TooManyFieldsSent" OR "PermissionDenied")
=== grep ===
grep -P '^(?:.*(?:.*SuspiciousOperation|.*DisallowedHost|.*DisallowedModelAdminLook
up|.*DisallowedModelAdminToField|.*DisallowedRedirect|.*InvalidSessionKey|.*Request
DataTooBig|.*SuspiciousFileOperation|.*SuspiciousMultipartForm|.*SuspiciousSession|
.*TooManyFieldsSent|.*PermissionDenied))'
# 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
```

# Example: Spring Framework Exceptions

```
title: Spring framework exceptions
description: Detects suspicious Spring framework exceptions that could indicate exploitation attempts
author: Thomas Patzke
reference:
  - https://docs.spring.io/spring-security/site/docs/current/apidocs/overview-tree.html
logsource:
  category: application
  product: spring
detection:
  keywords:
    - AccessDeniedException
    - CsrfException
    - InvalidCsrfTokenException
    - MissingCsrfTokenException
    - CookieTheftException
    - InvalidCookieException
    - RequestRejectedException
  condition: keywords
falsepositives:
  - Application bugs
  - Penetration testing
level: medium
```

# Example:

## Python PEP249 Exceptions

```
title: Python SQL Exceptions
description: Generic rule for SQL exceptions in Python according to PEP 249
author: Thomas Patzke
reference:
  - https://www.python.org/dev/peps/pep-0249/#exceptions
logsource:
  category: application
  product: python
detection:
  exceptions:
    - DataError
    - IntegrityError
    - ProgrammingError
    - OperationalError
  condition: exceptions
falsepositives:
  - Application bugs
  - Penetration testing
level: medium
```

# Example:

# OpenSSH Error Messages

```
title: Suspicious SSHD error
description: Detects suspicious SSH / SSHD error messages that indicate a fatal or suspicious error that could be caused by exploiting attempts
reference: https://github.com/openssh/openssh-portable/blob/master/ssherr.c
author: Florian Roth
date: 2017/06/30
logsource:
  product: linux
  service: sshd
detection:
  keywords:
    - 'unexpected internal error'
    - 'unknown or unsupported key type'
    - 'invalid certificate signing key'
    - 'invalid elliptic curve value'
    - 'incorrect signature'
    - 'error in libcrypto'
    - 'unexpected bytes remain after decoding'
  condition: keywords
falsepositives:
  - Unknown
level: medium
```

# 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:

## 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!2843&
logsource:
  product: windows
  service: sysmon
detection:
  selection:
    - EventID: 10
      TargetImage: 'C:\windows\system32\lsass.exe'
      GrantedAccess: '0x1410'
  condition: selection
falsepositives:
  - unknown
level: high
```

```
=== es-qs ===
(EventID:"10" AND GrantedAccess:"0x1410" AND TargetImage:"C:\\windows\\system32\\lsass.exe")
=== splunk ===
(EventID="10" GrantedAccess="0x1410" TargetImage="C:\\windows\\system32\\lsass.exe")
=== logpoint ===
(GrantedAccess="0x1410" TargetImage="C:\\windows\\system32\\lsass.exe" EventID="10")
```

# **Sigma Converter**



# Sigma Converter

Conversion of a Sigma rule into three different query languages:

- Splunk
- Elasticsearch
- LogPoint

# Sigma Converter

Conversion of a Sigma rule into three different query languages:

- Splunk
- Elasticsearch
- LogPoint

```
$ tools/sigmac.py -t splunk rules/windows/sysmon/sysmon_webshell_detection.yml
((ParentImage="*\\apache*" OR ParentImage="*\\tomcat*" OR ParentImage="*\\w3wp.exe" OR ParentImage="*\\php-cgi.exe"
OR ParentImage="*\\nginx.exe" OR ParentImage="*\\httpd.exe") (CommandLine="whoami" OR CommandLine="net user"
OR CommandLine="ping -n" OR CommandLine="systeminfo") EventID="1")
```

```
$ tools/sigmac.py -t es-qs rules/windows/sysmon/sysmon_webshell_detection.yml
(EventID:"1" AND CommandLine:("whoami" "net user" "ping \-n" "systeminfo")
AND ParentImage:("*\\apache*" "*\\tomcat*" "*\\w3wp.exe" "*\\php\ -cgi.exe" "*\\nginx.exe" "*\\httpd.exe"))
```

```
$ tools/sigmac.py -t logpoint rules/windows/sysmon/sysmon_webshell_detection.yml
(ParentImage IN ["*\\apache*", "*\\tomcat*", "*\\w3wp.exe", "*\\php-cgi.exe", "*\\nginx.exe", "*\\httpd.exe"]
EventID="1" CommandLine IN ["whoami", "net user", "ping -n", "systeminfo"])
```

# 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/sigmac.py -t splunk rules/windows/sysmon/sysmon_webshell_detection.yml
((ParentImage="*\\apache*" OR ParentImage="*\\tomcat*" OR ParentImage="*\\w3wp.exe" OR ParentImage="*\\php-cgi.exe"
OR ParentImage="*\\nginx.exe" OR ParentImage="*\\httpd.exe") (CommandLine="whoami" OR CommandLine="net user"
OR CommandLine="ping -n" OR CommandLine="systeminfo") EventID="1")
```

```
$ tools/sigmac.py -t es-qs rules/windows/sysmon/sysmon_webshell_detection.yml
(EventID:"1" AND CommandLine:("whoami" "net user" "ping \-n" "systeminfo")
AND ParentImage:("*\\apache*" "*\\tomcat*" "*\\w3wp.exe" "*\\php\-cgi.exe" "*\\nginx.exe" "*\\httpd.exe"))
```

```
$ tools/sigmac.py -t logpoint rules/windows/sysmon/sysmon_webshell_detection.yml
(ParentImage IN ["*\\apache*", "*\\tomcat*", "*\\w3wp.exe", "*\\php-cgi.exe", "*\\nginx.exe", "*\\httpd.exe"]
EventID="1" CommandLine IN ["whoami", "net user", "ping -n", "systeminfo"])
```

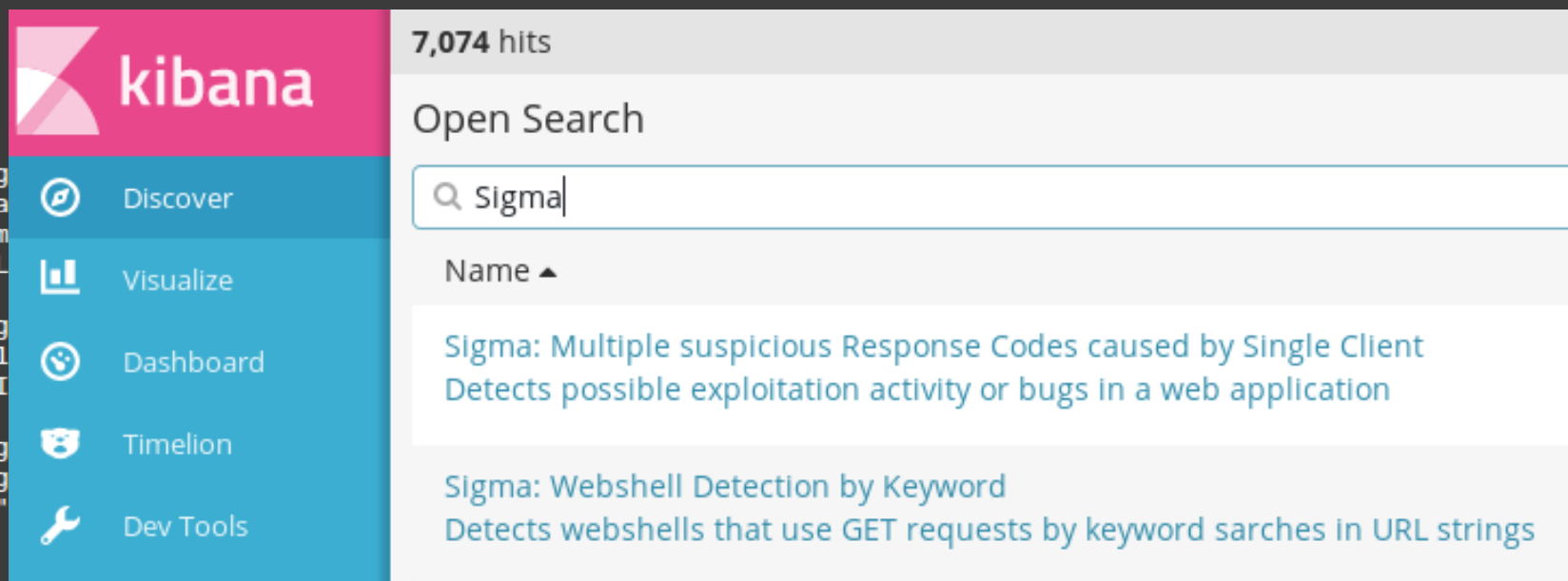
# Sigma Converter

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

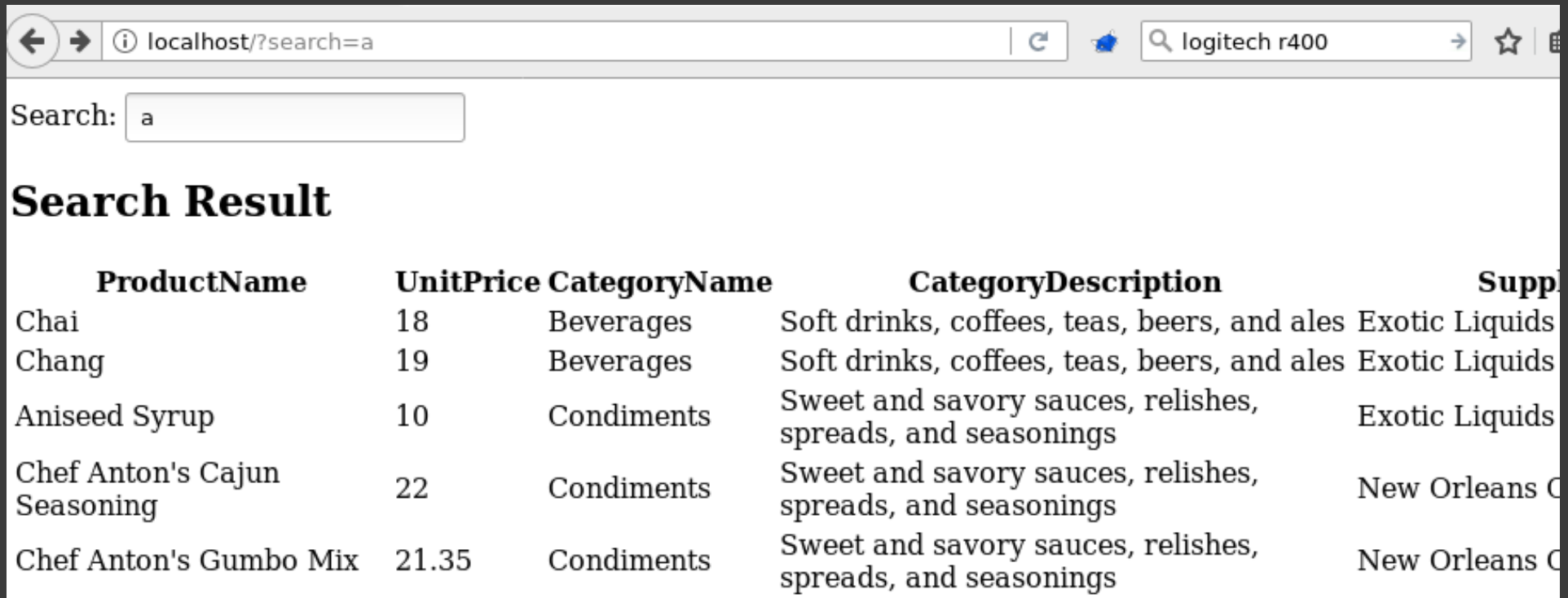


The screenshot shows the Kibana search interface. On the left is a sidebar with the Kibana logo and navigation options: Discover, Visualize, Dashboard, Timelion, and Dev Tools. The main content area shows a search for "Sigma" with 7,074 hits. Below the search bar, there are two search results listed:

- 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 searches in URL strings

**Demo Time!**

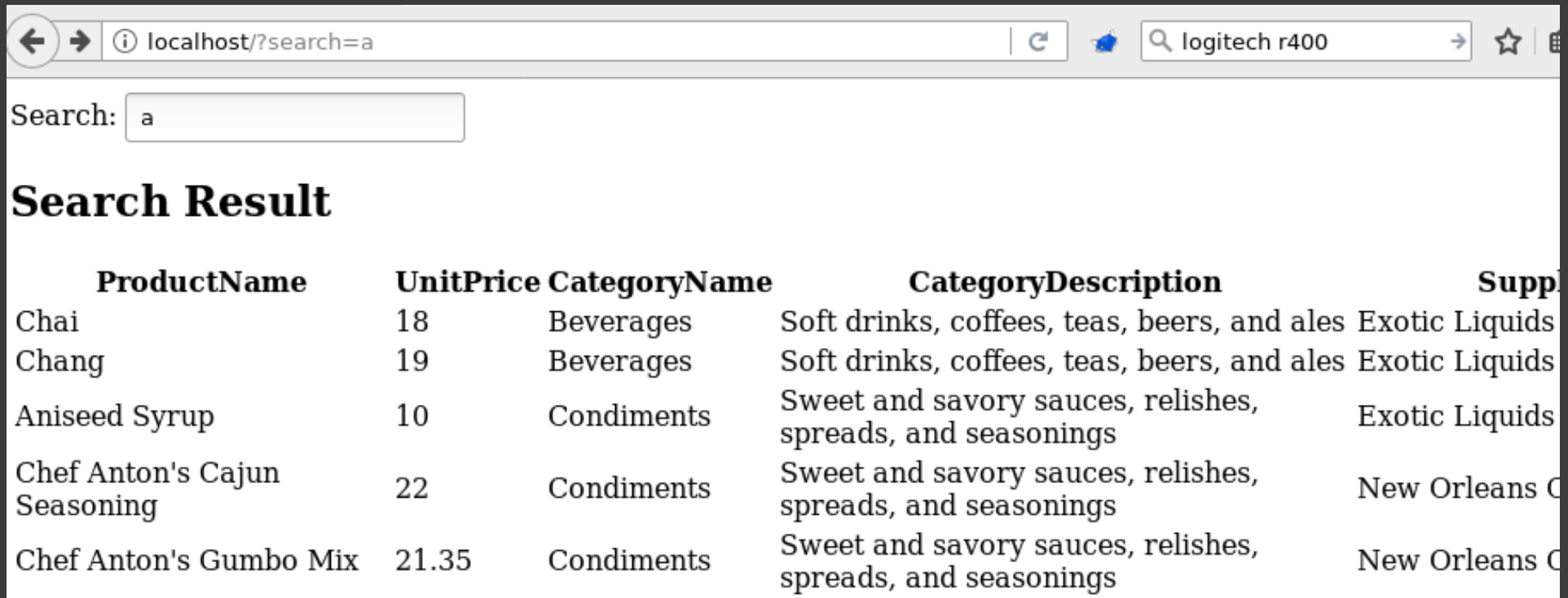
# Demo Time!



The screenshot shows a web browser window with the address bar containing 'localhost/?search=a'. The search bar contains the letter 'a'. Below the search bar, the heading 'Search Result' is displayed. A table of search results follows, with columns for ProductName, UnitPrice, CategoryName, CategoryDescription, and SupplierName. The results include Chai, Chang, Aniseed Syrup, Chef Anton's Cajun Seasoning, and Chef Anton's Gumbo Mix.

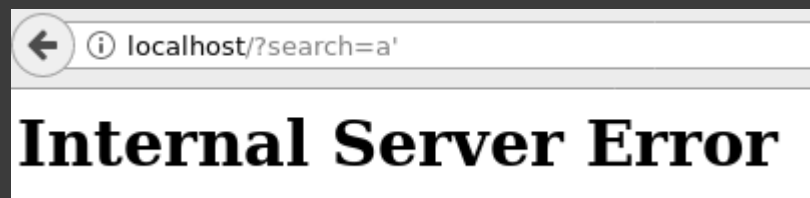
ProductName	UnitPrice	CategoryName	CategoryDescription	SupplierName
Chai	18	Beverages	Soft drinks, coffees, teas, beers, and ales	Exotic Liquids
Chang	19	Beverages	Soft drinks, coffees, teas, beers, and ales	Exotic Liquids
Aniseed Syrup	10	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings	Exotic Liquids
Chef Anton's Cajun Seasoning	22	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings	New Orleans C
Chef Anton's Gumbo Mix	21.35	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings	New Orleans C

# Demo Time!



A screenshot of a web browser window. The address bar shows 'localhost/?search=a'. The search bar contains the letter 'a'. Below the search bar, the heading 'Search Result' is displayed. A table of search results follows, with columns for ProductName, UnitPrice, CategoryName, CategoryDescription, and SupplierName. The results include Chai, Chang, Aniseed Syrup, Chef Anton's Cajun Seasoning, and Chef Anton's Gumbo Mix.

ProductName	UnitPrice	CategoryName	CategoryDescription	SupplierName
Chai	18	Beverages	Soft drinks, coffees, teas, beers, and ales	Exotic Liquids
Chang	19	Beverages	Soft drinks, coffees, teas, beers, and ales	Exotic Liquids
Aniseed Syrup	10	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings	Exotic Liquids
Chef Anton's Cajun Seasoning	22	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings	New Orleans C
Chef Anton's Gumbo Mix	21.35	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings	New Orleans C



A screenshot of a web browser window showing an error. The address bar shows 'localhost/?search=a'. Below the address bar, the heading 'Internal Server Error' is displayed in a large, bold font.

## Internal Server Error

201 hits

New Save Open Share < Last 15 minutes >

Search... (e.g. status:200 AND extension:PHP)

Uses lucene query syntax



Add a filter +

filebeat-\*

Selected Fields

? \_source

Available Fields



@timestamp

t \_id

t \_index

# \_score

t \_type

t beat.hostname

t beat.name

t beat.version

t input\_type

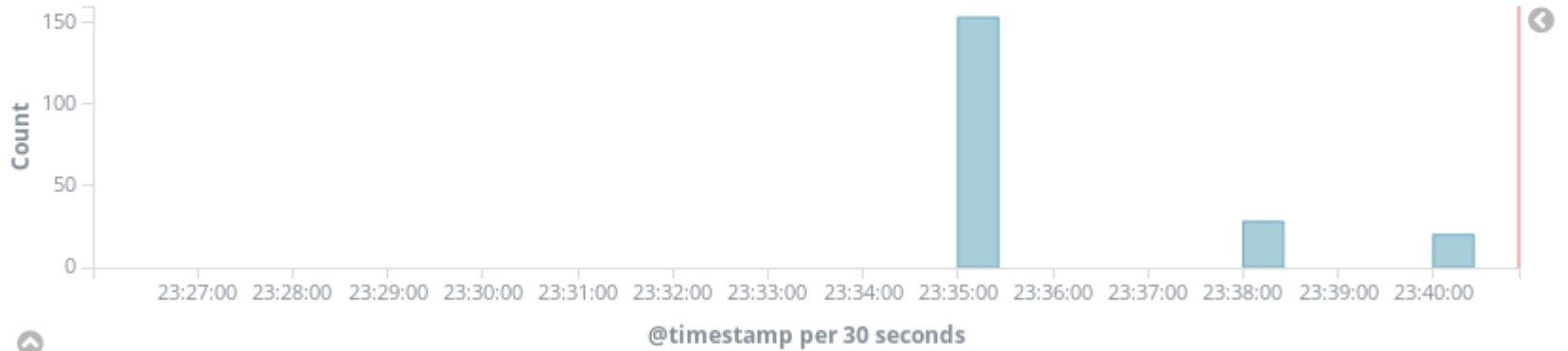
t message

# offset

t source

t type

November 13th 2017, 23:25:54.145 - November 13th 2017, 23:40:54.145 — Auto



Time	_source
▶ November 13th 2017, 23:40:06.758	<code>@timestamp:</code> November 13th 2017, 23:40:06.758 <code>beat.hostname:</code> 8f223273506a <code>beat.name:</code> 8f223273506a <code>beat.version:</code> 5.6.3 <code>input_type:</code> log <code>message:</code> [Mon Nov 13 22:40:06.363903 2017] [wsgi:error] [pid 11:tid 140352891807488] [remote 172.20.0.1:16754] File "/usr/local/lib/python3.5/dist-packages/flask/app.py", line 1614, in full_dispatch_request <code>offset:</code> 17,639 <code>source:</code> /var/log/apache2
▶ November 13th 2017, 23:40:06.758	<code>@timestamp:</code> November 13th 2017, 23:40:06.758 <code>beat.hostname:</code> 8f223273506a <code>beat.name:</code> 8f223273506a <code>beat.version:</code> 5.6.3 <code>input_type:</code> log <code>message:</code> [Mon Nov 13 22:40:06.363906 2017] [wsgi:error] [pid 11:tid 140352891807488] [remote 172.20.0.1:16754] rv = self.handle_user_exception(e) <code>offset:</code> 17,780 <code>source:</code> /var/log/apache2/vulnerable-error.log <code>type:</code> log <code>_id:</code> AV-3ijBOKlw1vZjwWPX6
▶ November 13th 2017, 23:40:06.758	<code>@timestamp:</code> November 13th 2017, 23:40:06.758 <code>beat.hostname:</code> 8f223273506a <code>beat.name:</code> 8f223273506a <code>beat.version:</code> 5.6.3 <code>input_type:</code> log <code>message:</code> [Mon Nov 13 22:40:06.363916 2017] [wsgi:error] [pid 11:tid 140352891807488] [remote





201 hits

New Save Open Share < Last 15 minutes >

Search... (e.g. status:200 AND extension:PHP)

Uses lucene query syntax



Add a filter +

filebeat-\*

November 13th 2017, 23:25:54.145 - November 13th 2017, 23:40:54.145 —

Auto

Selected Fields

? \_source

Available Fields

@timestamp

t \_id

t \_index

# \_score

t \_type

t beat.hostname

t beat.name

t beat.version

t input\_type

t message

# offset

t source

t type

```
$ tools/sigmac.py -t kibana -c tools/config/elk-defaultindex-filebeat.yml -r rules/application
[
  {
    "_type": "search",
    "_source": {
      "version": 1,
      "sort": [
```

201 hits

New Save Open

### Open Search

Q Saved Searches Filter...

Name ▲

Django framework exceptions

Detects suspicious Django web application framework exceptions that could indicate exploitation attempts

Python SQL Exceptions

Generic rule for SQL exceptions in Python according to PEP 249

Ruby on Rails framework exceptions

Detects suspicious Ruby on Rails exceptions that could indicate exploitation attempts

Spring framework exceptions

Detects suspicious Spring framework exceptions that could indicate exploitation attempts

23273506a b

sage: [Mon N

488] [remote

ask/app.py",

/log/apache2

23273506a b

sage: [Mon N

488] [remote

780 source

IKlw1vZjwWPX6

23273506a b

eat.name: 8f223273506a beat.version: 5.6.3 input\_type: log message: [Mon N

ov 13 22:40:06.363916 2017] [wsgi:error] [pid 11:tid 140352891807488] [remote

201 hits

New Save Open Share < Last 15 minutes >

Python SQL Exceptions 6 hits

New Save Open Share < Last 15 minutes >

("DataError" OR "IntegrityError" OR "ProgrammingError" OR "OperationalError")

Uses lucene query syntax

Add a filter +

filebeat-\*

Selected Fields

? \_source

Available Fields

@timestamp

t \_id

t \_index

# \_score

t \_type

t beat.hostname

t beat.name

t beat.version

t input\_type

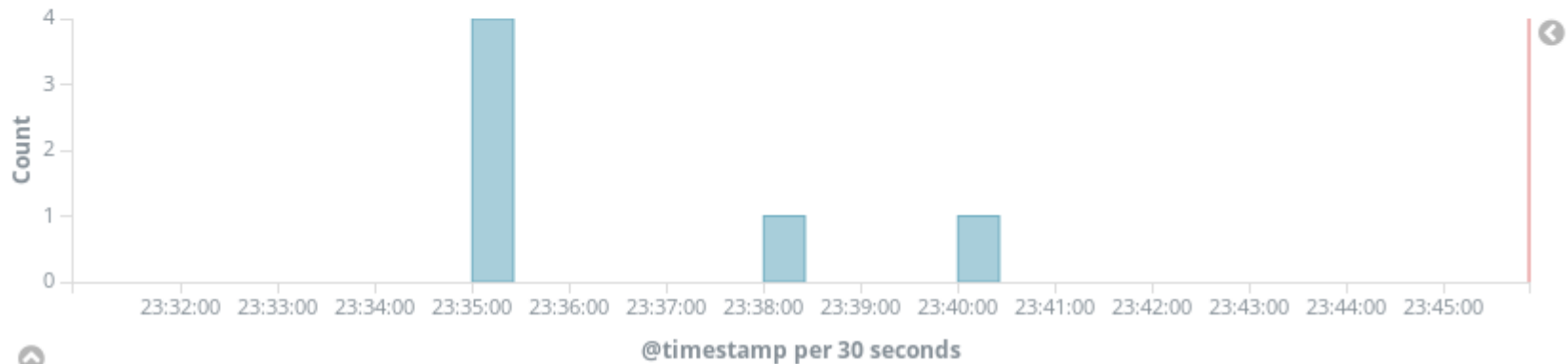
t message

# offset

t source

November 13th 2017, 23:30:52.800 - November 13th 2017, 23:45:52.800

Auto



Time	_source
▶ November 13th 2017, 23:40:06.758	<pre>message: [Mon Nov 13 22:40:06.363944 2017] [wsgi:error] [pid 11:tid 140352891807488] [remote 172.20.0.1:16754] sqlite3.OperationalError: unrecognized token: "" @timestamp: November 13th 2017, 23:40:06.758 beat.hostname: 8f223273506a beat.name: 8f223273506a beat.version: 5.6.3 input_type: log offset: 19,726 source: /var/log/apache2/vulnerable-error.log type: log _id: AV-3</pre>
▶ November 13th 2017, 23:38:21.751	<pre>message: [Mon Nov 13 22:38:20.524281 2017] [wsgi:error] [pid 11:tid 140352942171904] [remote 172.20.0.1:16754] sqlite3.OperationalError: unrecognized token: "" @timestamp: November 13th 2017, 23:38:21.751 beat.hostname: 8f223273506a beat.name: 8f223273506a beat.version: 5.6.3 input_type: log offset: 16,617 source: /var/log/apache2/vulnerable-error.log type: log _id: AV-3 eat.name: 8f223273506a beat.version: 5.6.3 input_type: log message: [Mon Nov 13 22:40:06.363916 2017] [wsgi:error] [pid 11:tid 140352891807488] [remote</pre>

# Challenges in Rule Conversion

# Challenges in Rule Conversion

- Usage of different field names
  - Solution: field name mappings from Sigma rule field names to SIEM/environment specific names

# 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

# 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

# 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



# 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

# Current State & Future

- Rules
  - Many rules for Windows/OS and network events
  - Few application rules, room for improvement!
- Sigma Converter
  - Some backends, but more required
  - Further improvements
- Further tool ideas:
  - Sigma Rule Builder Webapp
  - Automatic rule building from log samples

**How can you Contribute?**

# How can you Contribute?

- Developers
  - Log verbosely!
    - Access check violations
    - Failing security checks (wrong CSRF token, ...)
    - If requests are wrong (too many, too few parameters, wrong value types, ...)
    - Unexpected states (skipped workflow steps, ...)
  - Provide Sigma rules with your project




# How can you Contribute?

- Developers
  - Log verbosely!
    - Access check violations
    - Failing security checks (wrong CSRF token, ...)
    - If requests are wrong (too many, too few parameters, wrong value types, ...)
    - Unexpected states (skipped workflow steps, ...)
  - Provide Sigma rules with your project
- Pentesters & Security Researchers
  - Check logs for attack traces and build Sigma rules

# How can you Contribute?

- Developers
  - Log verbosely!
    - Access check violations
    - Failing security checks (wrong CSRF token, ...)
    - If requests are wrong (too many, too few parameters, wrong value types, ...)
    - Unexpected states (skipped workflow steps, ...)
  - Provide Sigma rules with your project
- Pentesters & Security Researchers
  - Check logs for attack traces and build Sigma rules
- Code/Tool contributions are always welcome!

# Questions?

- Rules + Code: <https://github.com/Neo23x0/sigma>
- Documentation: <https://github.com/Neo23x0/sigma/wiki>
- Thomas Patzke
  -  @blubbfiction
  -  thomas@patzke.org
- Florian Roth
  -  @cyb3rops
- <https://www.bsk-consulting.de/2017/07/06/the-best-possible-monitoring-with-sigma-rules/>