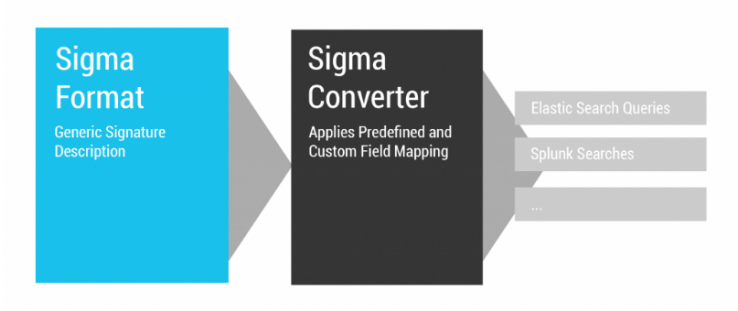
Sigma Rules

Sharing SIEM rules can be an extremely beneficial process for a security team, whether they're sharing them or retrieving them, but SIEM rules are written in specific structures depending on the SIEM platform. While it's possible to share the logic of the rule (how it works) in plain English, there is a better way to quickly share or ingest SIEM rules shared by teams around the world. In this lesson we're going to introduce you to Sigma.

To ensure accuracy we've copied information directly from the Sigma Github page available here.

What is Sigma?

Sigma is a generic and open signature format that allows you to describe relevant log events in a straightforward manner. The rule format is very flexible, easy to write and applicable to any type of log file. The main purpose of this project is to provide a structured form in which researchers or analysts can describe their once developed detection methods and make them shareable with others.



https://github.com/Neo23x0/sigma

Rules can be written in the Sigma language and then using a converter (Sigmac) they can be exported as rules in the correct format for a number of different SIEM platforms. This process can also be reversed allowing security professionals to export rules from their vendor format to Sigma format so they can be used by teams with a different SIEM.

Which Platforms Support Sigma?

- Splunk
- QRadar
- ArcSight
- Elasticsearch (Elastalert, Query strings, DSL, Watcher, & Kibana)
- Logpoint

Benefits of Using Sigma

- Describe your detection method in Sigma to make it sharable
- Write your SIEM searches in Sigma to avoid a vendor lock-in (meaning you can flexibly change SIEM solution without having to lose all of your custom rules)
- Share the signature in the appendix of your analysis or research report along with IOCs and YARA rules to

- Share the signature in the appendix of your analysis or research report along with IOCs and YARA rules to allow others to replicate your work and build detection rules
- Share the signature in threat intel communities (ISACs) – e.g. via MISP (which we covered in the Threat Intel domain!)

SIGMA Rule Example

In this example we're looking at a Sigma rule that can detect when a web server has been compromised and is running a web shell, allowing a malicious actor to visit a specific URL which will provide them with a console, allowing them to execute commands as if they are on the server. We'll break down the rule below, even though it's really human-readable!

web_webshell_keyword.yml | win_alert_mimikatz_keywords.yml | win_susp_eventlog_cleared.yml

```
1  title: Webshell Detection by Keyword
2  description: Detects webshells that use GET requests by keyword sarches in URL strings
3  author: Florian Roth
4  logsource:
5      type: webserver
6  detection:
7      keywords:
8          - '=whoami'
9          - '=net%20user'
10         - '=cmd%20/c%20'
11     condition: selection and keywords
12 falsepositives:
13     - Web sites like wikis with articles on os commands and pages that include the os commands in the
       URLs
14     - User searches in search boxes of the respective website
15 level: high
16
```

On line 6 we can see that the 'detection' is declared, stating how this rule works. Line 7 states it is using the method of matching keywords against a URL string (mentioned don Line 2).

So if this rule was actively being used to monitor a web server and we had a web shell running on *https://example.com/13919595/asjkdasjdkasjvn/shell.php?*, and we visited the interface and tried to use the command 'whoami' this would be included in a POST HTTP request to the web server, meaning the URL will be changed to include '=whoami'. This activity would generate an alert for the security team to investigate, making them aware of the web shell. It is extremely unlikely that a normal visitor would ever need to include these operating system commands in a POST URL request so there is a low rate of false positives (but some scenarios are covered on lines 13 and 14).

There are some great real-world rules to take a look at on Florian Roth's Github page GitHub – Neo23x0/sigma: Generic Signature Format for SIEM Systems. Additional rules are available at https://github.com/SigmaHQ/sigma/tree/master/rules. We highly recommend taking a look at these to better understand how they function.

Previous Topic | Mark Complete | Next Topic
Back to Lesson