Sigma Update: What's new in Sigma

Thomas Patzke, 25.10.2019, TLP:White

How is Sigma related to ATT&CK?

~80% of Sigma Rules contained in the Sigma main repository have ATT&CK tags

What’s new in Sigma and Plans

1. Value modifiers
2. Improved usability of configurations
3. Rule licensing plans
4. Many small improvements and bug fixes

Value Modifiers

Motivation:

1. Writing encoded values in Sigma rules doesn't increases readability
2. Problematic encodings like UTF16
3. Changing default behaviors (e.g. AND of list elements) required workarounds

Solution: Value Modifiers

4. Post-processing of values
   - Encoding
   - Different logical linking of values
   - Treat as regular expression
5. Syntax:

```
field name|modifier1|modifier2|...
```

Implemented Value Modifiers

```
Modifiers:
  contains : Add *-wildcard before and after all string(s)
       all : Override default OR-linking behavior for list with AND-linking of all list values
    base64 : Encode strings with Base64
base64offset : Encode string(s) with Base64 in all three possible shifted offsets
     utf16 : Encode string to UTF-16 byte sequence
   utf16le : Encode string to UTF-16 little endian byte sequence
      wide : Modifier 'wide' is an alias for the utf16le modifier.
   utf16be : Encode string to UTF-16 big endian byte sequence
        re : Treat value as regular expression
```

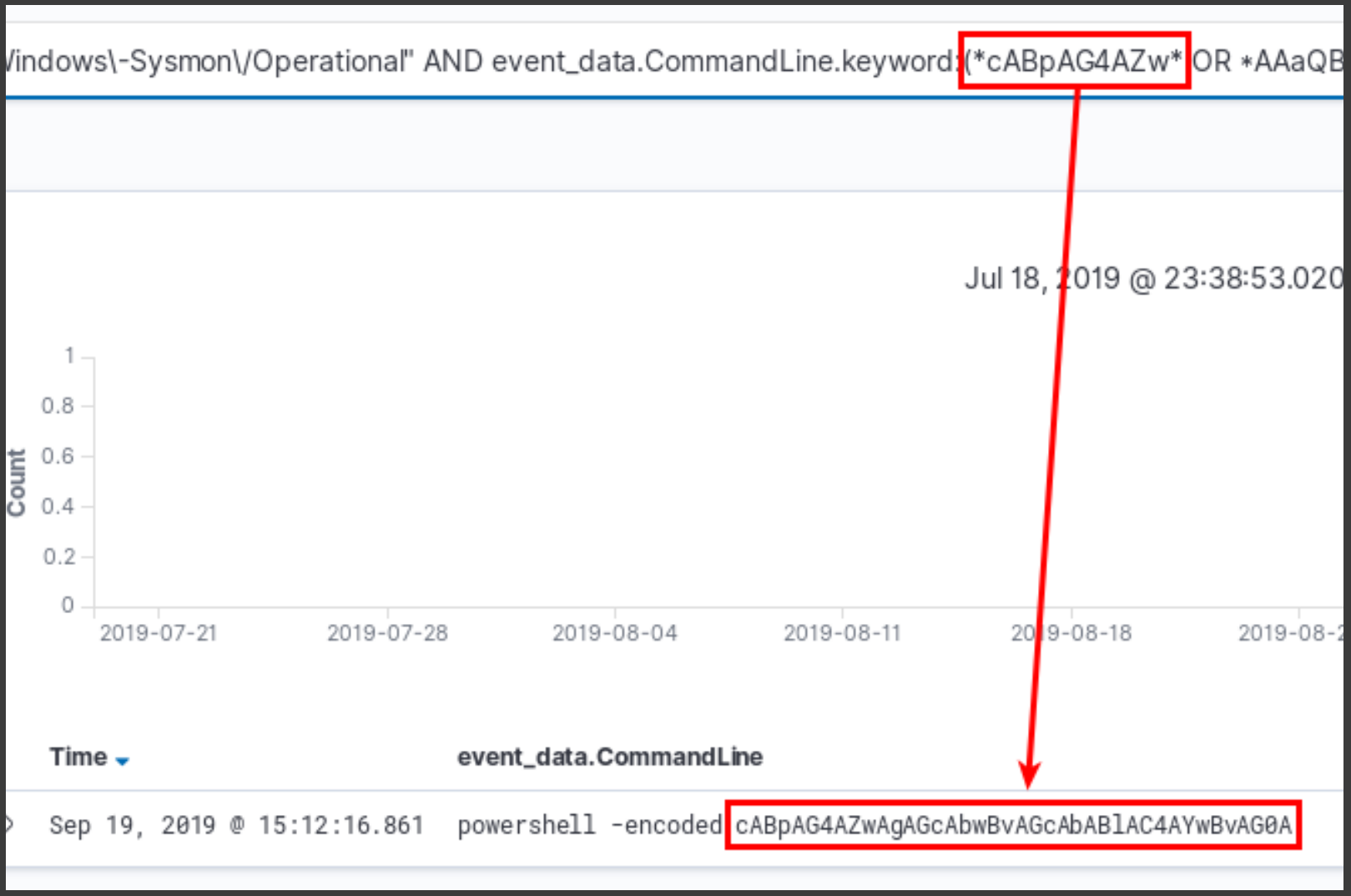
Value Modifiers: Sigma Rule

```yaml
title: Base-64 encoded ping
status: stable
description: Detect all possibilities of a base64 encoded ping command
tags:
    - attack.t1140
    - attack.t1027
    - attack.defense_evasion
logsource:
    category: process_creation
    product: windows
detection:
    selection:
        CommandLine|wide|base64offset|contains: 'ping'
    condition: selection
level: high
```

Value Modifiers: Generated Query

```
(event_id:"1"
  AND log_name:"Microsoft\-Windows\-
Sysmon\/Operational"
  AND event_data.CommandLine.keyword:
   (
     *cABpAG4AZw* OR
     *AaaQBuAGcA* OR
     *wAGkAbgBnA*
  ))
```

Value Modifiers: Result of Query



Value Modifiers: Decoded Result

Sigma Converter Configuration Problems

What was wrong with Sigma configurations?

- Usually they are required to generate proper queries, but most users weren't aware about this
- It was hard to determine, which config is appropriate
- Users had to enter the full paths of configurations
- It was possible to use stacked configurations in the wrong order
- Configuration could be used with wrong backends (e.g. Splunk backend with ES configuration)

Sigma Converter Configuration Improvements

- Sigma converter now refuses to convert rules if configuration is required or configuration doesn't fits to backend
- Converter suggests appropriate configurations
- Discovery of available configurations (`sigmac -l`)
- Every configuration has a title and a description
- Configurations can be used by specifying the filename without path or suffix
- Order of configurations is checked

```
Configurations:
                    sumologic : SumoLogic
          logstash-defaultindex : Generic Logstash index prefix
                     powershell : Logsource to LogName mappings for PowerShell backend
              logstash-windows : Logstash Windows common log sources
                splunk-windows : Splunk Windows log source conditions
           splunk-windows-index : Splunk Windows index and EventID field mapping
                     netwitness : NetWitness
```

```
[thomas:~/Devel/sigma] master+* ± sigmac -t es-qs -c splunk-windows rules/windows/process_creation/win_malware_emotet.yml
The configuration 'splunk-windows' is not valid for backend 'es-qs'. Valid choices are: splunk, splunkxml
```

```
[thomas:~/Devel/sigma] master+* 20 ± sigmac -t es-qs rules/windows/process_creation/win_malware_emotet.yml
The backend you want to use usually requires a configuration to generate valid results. Please provide one with --config/-c.
Available choices for this backend (get complete list with --lists/-l):
          logstash-defaultindex : Generic Logstash index prefix
               logstash-windows : Logstash Windows common log sources
                 logstash-linux : Logstash Linux project (https://github.com/thomaspatzke/logstash-linux)
    winlogbeat-modules-enabled : Elastic Winlogbeat (from 7.x) index pattern and field
```

Bug Fixes and Improvements

- Elastic stack 7.x support
- Many improvements and fixes of backend code
- New backends: SumoLogic, elastalert-dsl
- Code cleanup

Rule Licensing

- Currently rules from the open source repository are licensed with GPL that has several issues:
  - Is the generated query also GPL licensed?
  - If yes: must baselining be contributed back?
- Planned switch to other license: https://github.com/Neo23x0/sigma/issues/382
  - Allowance of commercial usage
  - Attribution is a **must**
  - CC BY-SA? MIT? …?
- Added a License attribute in Sigma rules

Plans

- Stable releases
- Improving coverage of Sigma ruleset (aka OSCD)
- Improvement of documentation
- Tool to convert ATT&CK mapping to new ATT&CK identifiers with sub-techniques
- Extend Sigma backends with query capabilities
- MISP2Sigma
- Detection testing framework
  - Does the detection identify the log events it should?
  - Testing if queries generated from Sigma rules are semantically correct
  - But also generic testing of queries against SIEM systems
  - Likely as separate project

The “Driving Home from hack.lu” Project

- Every time I drive home from hack.lu I develop a small project (last time: sigma2misp)
- This time: Sigma rule distance calculator
  - Compare Sigma rules and get a similarity score
  - Idea: normalizing a Sigma rule into one long string and calculate the Levenshtein distance

Open Security Collaborative Development

- https://oscd.community/
- Initiated by the community
- Thanks to hack.lu organizers to give us the room for the workshop
- Currently 74 (!) new rules