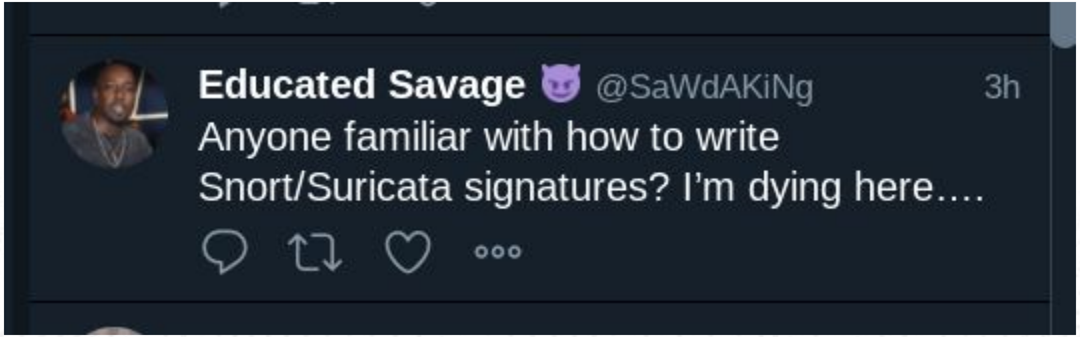
Threat detection engineering with Suricata

Building modern network detection with Suricata

Writing Suricata Signatures

Some dare to say out loud what people are thinking quietly



Educated Savage 😈 @SaWdAKiNg 3h

Anyone familiar with how to write Snort/Suricata signatures? I'm dying here….

Introduction

Short intro to the mob

Peter Manev & Eric Leblond

Peter Manev
@pevma
16 yrs with Suricata
OISF Exec team
Book author

Suricata Evangelist
QA/Training lead

CSO Stamus Networks
Clear NDR Community (SELKS) maintainer

Eric Leblond
@regit@infosec.exchange

CTO of Stamus Networks

16 yrs with Suricata
25 yrs of C dev

OISF Board of Directors
Book author
Emeritus member of Netfilter Coreteam

Stamus Networks

Headquarters: Indianapolis, USA and Paris, France

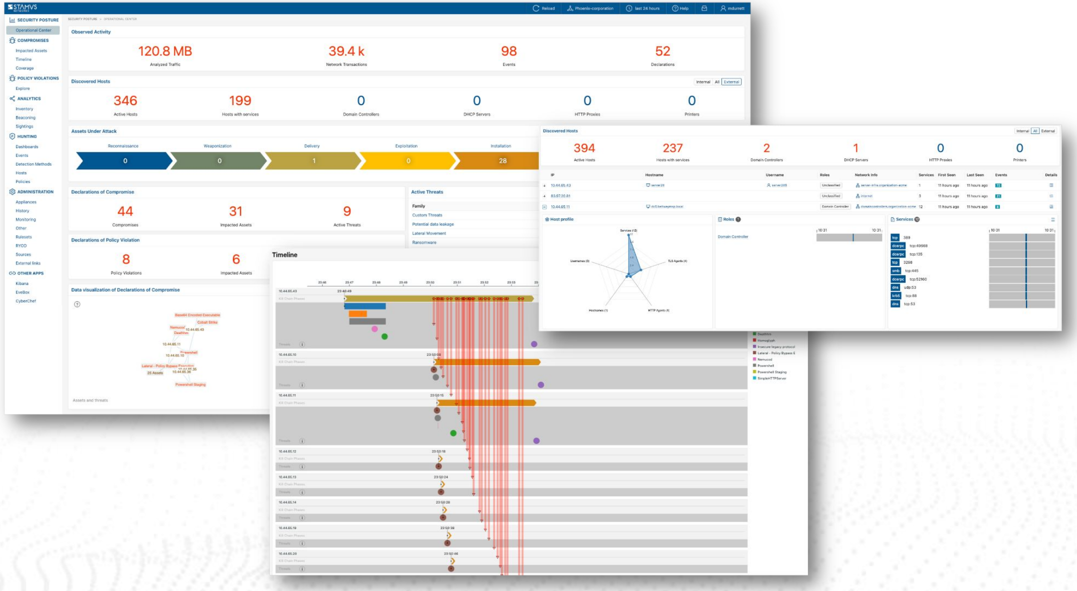
Team: Global team in 8 countries

Funding: Closed $6M Series A - August 2023

Clients: Global enterprises and institutions in 16 countries

Proven in NATO cyber exercises: 7+ years participation in CCDCOE Locked Shields and Crossed Swords

Editors of Clear NDR, a Suricata based NDR solution



Observed Activity

120.8 MB Analyzed Traffic

39.4 k Network Transactions

98 Events

52 Declarations

Discovered Hosts

346 Active Hosts

199 Hosts with services

0 Domain Controllers

0 DHCP Servers

0 HTTP Proxies

0 Printers

Declarations of Compromise

44 Compromises

31 Impacted Assets

9 Active Threats

Declarations of Policy Violation

8 Policy Violations

6 Impacted Assets

Discovered Hosts

394 Active Hosts

237 Hosts with services

2 Domain Controllers

1 DHCP Servers

0 HTTP Proxies

0 Printers

IP	Hostname	Username	Roles	Network Info	Services	First Seen	Last Seen	Events	Details
10.44.65.43	server28	server28$	Unclassified	server-infra.organization-acme	1	11 hours ago	11 hours ago	72	
83.97.20.81			Unclassified	internet	3	11 hours ago	11 hours ago	21	
10.44.65.11	dc5.beltwaymsp.local		Domain Controller	domaincontrollers.organization-acme	12	11 hours ago	11 hours ago	8	

tcp	389
dcerpc	tcp:49668
dcerpc	tcp:135
tcp	3268
smb	tcp:445
dcerpc	tcp:52160
dns	udp:53
krb5	tcp:88
dns	tcp:53

A Long History of Open Source Innovations

SURICATA

Clear NDR Community

Stamus Networks App for Splunk

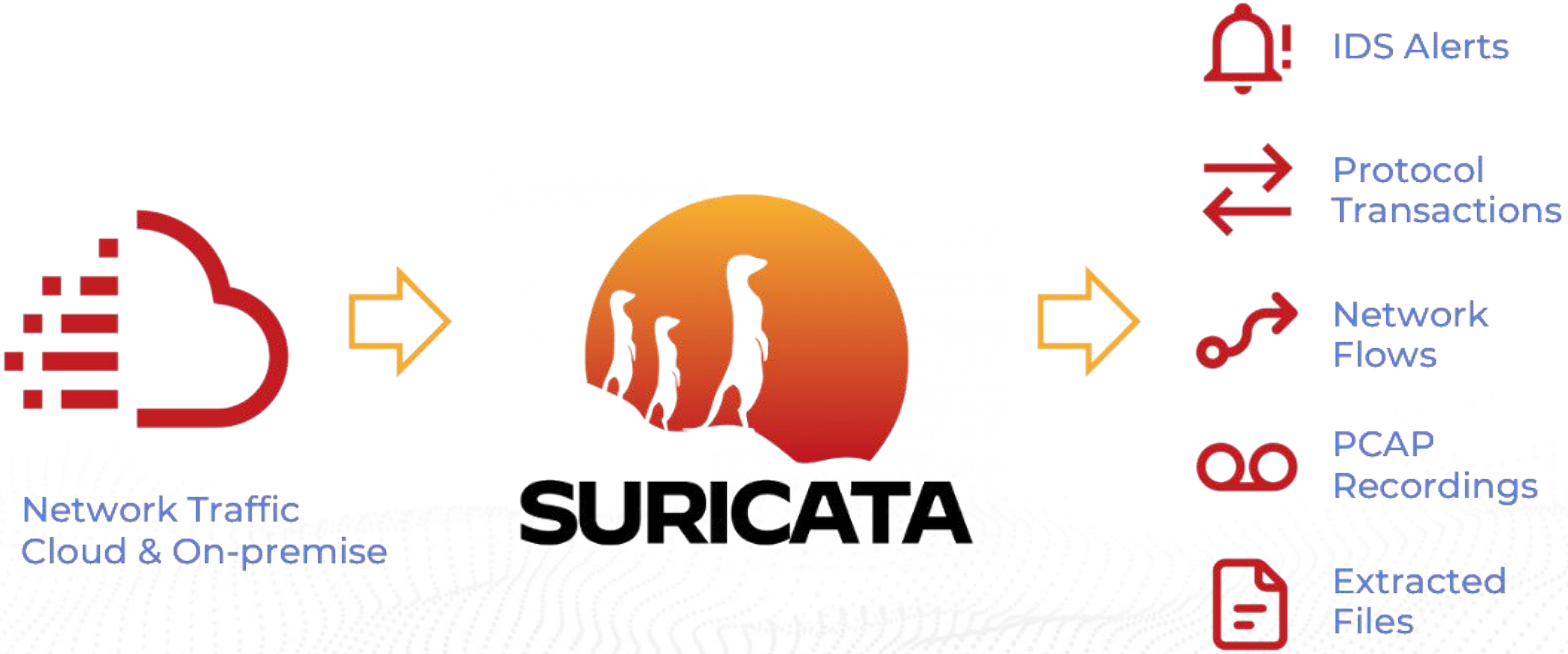
Jupyter Playbooks for Suricata

Lateral Movement Ruleset

Suricata Language Server

(Newly Registered Domains Ruleset)

Suricata is far more than an IDS/IPS



Network Traffic
Cloud & On-premise

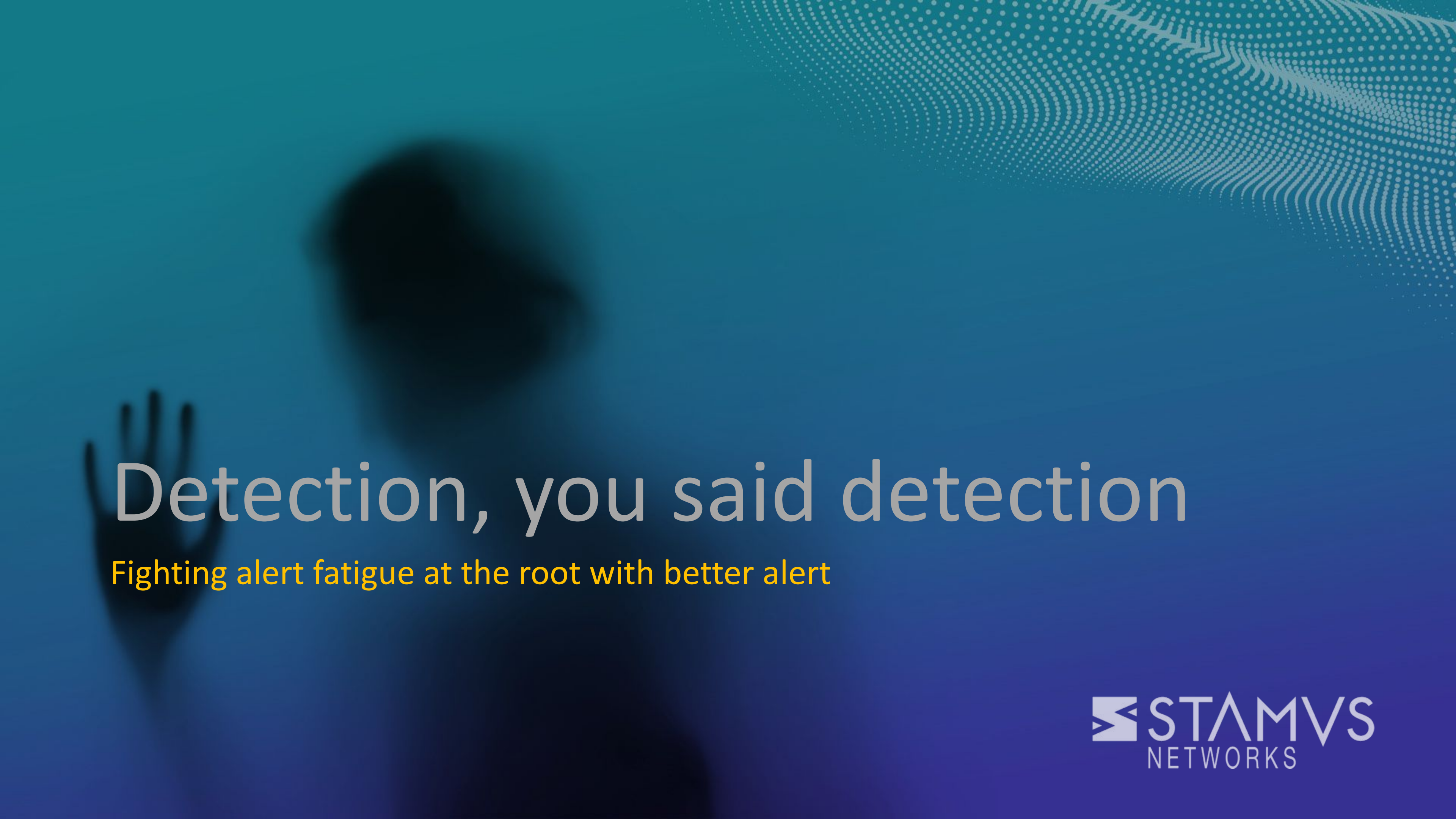
SURICATA

- IDS Alerts
- Protocol Transactions
- Network Flows
- PCAP Recordings
- Extracted Files

Source: Stamus Networks

Suricata

- Born: 2008
- Weight: 600 000 lines of code
- Composition: C, Rust
- Eat: live packets and dead ones
- Produce: JSON files/output
  - Protocol transaction
  - IDS alerts
  - PCAP
- Characteristics:
  - High speed
  - Open Source
  - Community driven
  - World famous
- Software owned and managed by the Open Information Security Foundation



Detection, you said detection

Fighting alert fatigue at the root with better alert

STAMVS NETWORKS

What is a good detection event ?

- Having all context possible
  - In the alert event
  - WITHOUT the need of any correlation to other events
  - But with the capability to correlate with other events
- Context must contain
  - Session and protocol information
    - What is known about the flow
    - What is known about the application layer
  - Information coming from rules engineering
    - Why detection was developed
    - What is its purpose
    - How is it classified

Suricata Installation

Let’s start hands on

STAMUS NETWORKS

Installation of Suricata 8.0.1

- Windows: https://letsdefend.io/blog/how-to-install-and-configure-suricata-on-windows
- Linux:
  - Use package manager
  - Change distribution if not available
- MacOS X:
  - brew install suricata

Caution Live Malware!

We will be reviewing in a lot of cases malware that can still be live or active!

Please DO NOT execute or trigger any of the binaries and / or visit any of the IoCs (domains/urls etc) anywhere else but a sandboxed environment.

Hands on + demo: Read a PCAP

- Get pcap from Malware Traffic Analysis
  - https://www.malware-traffic-analysis.net/2025/10/08/index.html
- Download rule file from ET Open:
  - https://rules.emergingthreats.net/open/suricata-7.0.3/emerging-all.rules
- Run suricata on the pcap loading the file
  - Create dir for output: mkdir out
  - Run Suricata: suricata -r FILE.pcap -S etopen.rules -l out -v
- Check eve.json file
  - How many alerts ?
  - How many events ?
  - Do you need Zeek ?

Rules structure

Signature format 101

Example:

```
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"HTTP GET Request Containing Rule in
URI"; flow:established,to server; http.method; content:"GET"; http.uri; content:"rule";
classtype:bad-unknown; sid:123; rev:1;)
```

Keywords:

- Action: alert
- Protocol: http
- Sticky buffers: http.method, http.uri
- Information: classtype
- Identifier: sid
- Revision: rev

Sid & Reference keyword

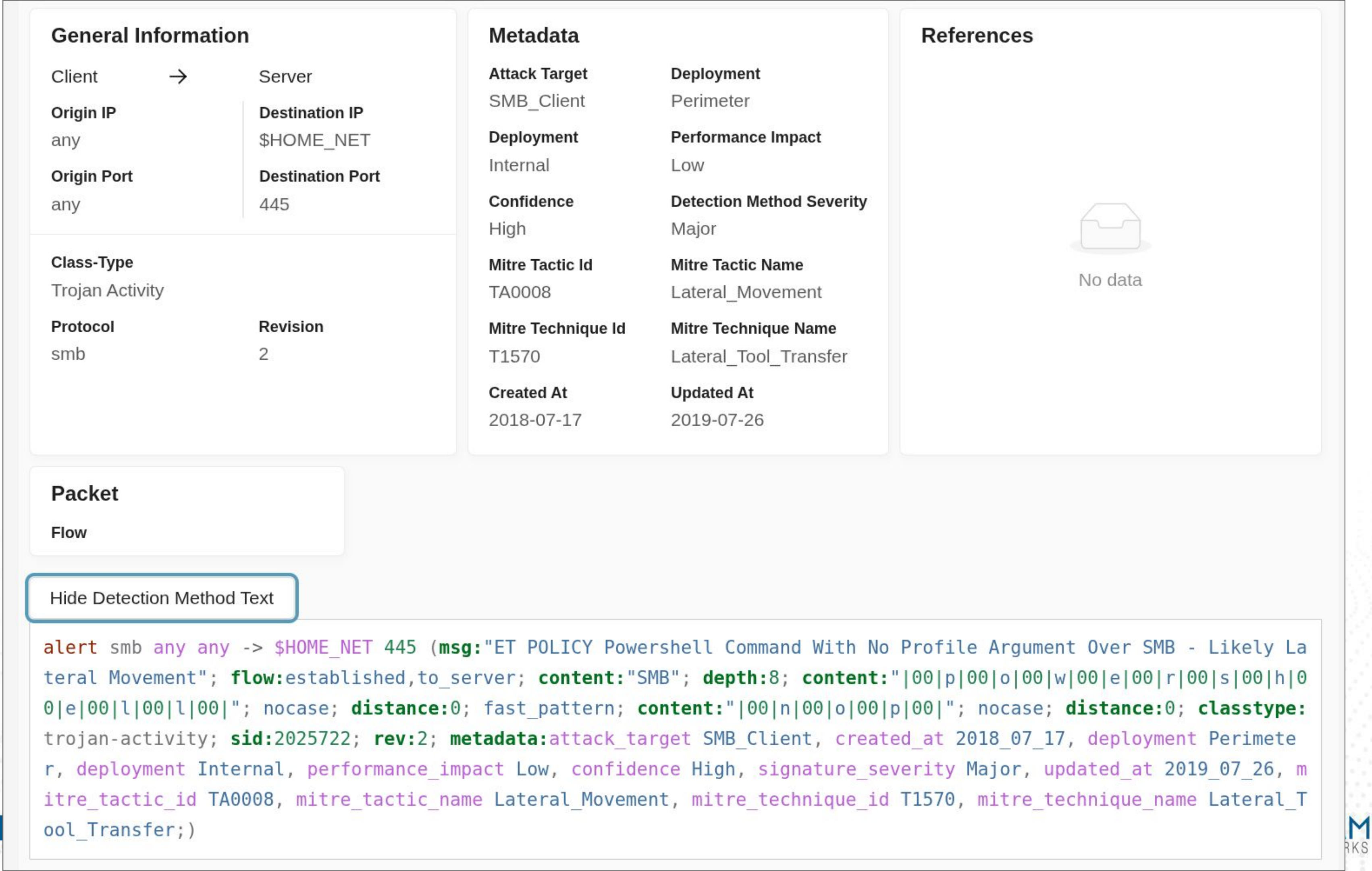
- sid: signature identifier
  - https://sidallocation.org/
  - Range 1000000-1999999 for private use
  - Ask your range if you want to publish
- reference:
  - Give context
  - URL to documentation
  - Example: reference:url,https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-scmr/0d7a7011-9f41-470d-ad52-8535b47ac282;

Metadata keyword

- Metadata information on signature
  - Arbitrary key value
  - Data ends up in the alert
- Used to transfert context

Using metadata

- This is convention
  - Mainly defined and used by Proofpoint
- Edition information
  - Creation date: created_at
  - Update date: updated_at
- Severity and target
  - Severity: signature_severity
  - Target: attack_target
- MITRE information
  - mitre_tactid_id, mitre_tactic_name
  - mitre_technique_id, mitre_technique_name



General Information

Client → Server

Origin IP any	**Destination IP** $HOME_NET
Origin Port any	**Destination Port** 445

Class-Type
Trojan Activity

Protocol smb	**Revision** 2

Metadata

Attack Target SMB_Client	**Deployment** Perimeter
Deployment Internal	**Performance Impact** Low
Confidence High	**Detection Method Severity** Major
Mitre Tactic Id TA0008	**Mitre Tactic Name** Lateral_Movement
Mitre Technique Id T1570	**Mitre Technique Name** Lateral_Tool_Transfer
Created At 2018-07-17	**Updated At** 2019-07-26

References

No data

Packet

Flow

```
alert smb any any -> $HOME_NET 445 (msg:"ET POLICY Powershell Command With No Profile Argument Over SMB - Likely Lateral Movement"; flow:established,to_server; content:"SMB"; depth:8; content:"|00|p|00|o|00|w|00|e|00|r|00|s|00|h|00|e|00|l|00|l|00|"; nocase; distance:0; fast_pattern; content:"|00|n|00|o|00|p|00|"; nocase; distance:0; classtype:trojan-activity; sid:2025722; rev:2; metadata:attack_target SMB_Client, created_at 2018_07_17, deployment Perimeter, deployment Internal, performance_impact Low, confidence High, signature_severity Major, updated_at 2019_07_26, mitre_tactic_id TA0008, mitre_tactic_name Lateral_Movement, mitre_technique_id T1570, mitre_technique_name Lateral_Tool_Transfer;)
```

Demo: Replay on Clear NDR Community

- Let’s replay:
  - stamusctl readpcap command
- Check for
  - metadata
  - reference

Hands-on: test output

- Write a signature
  - on hostname bryncoed.com on TLS transaction
    - https://docs.suricata.io/en/suricata-8.0.1/rules/index.html for keywords doc
  - Set metadata
    - author
    - created_at
    - updated_at
- Test the rule syntax with:
  - suricata -T -S my.rules
- Replay pcap loading the rules file
  - Use -S option to your file
- Check the alert in output directory

IOC Matching

STAMUS NETWORKS

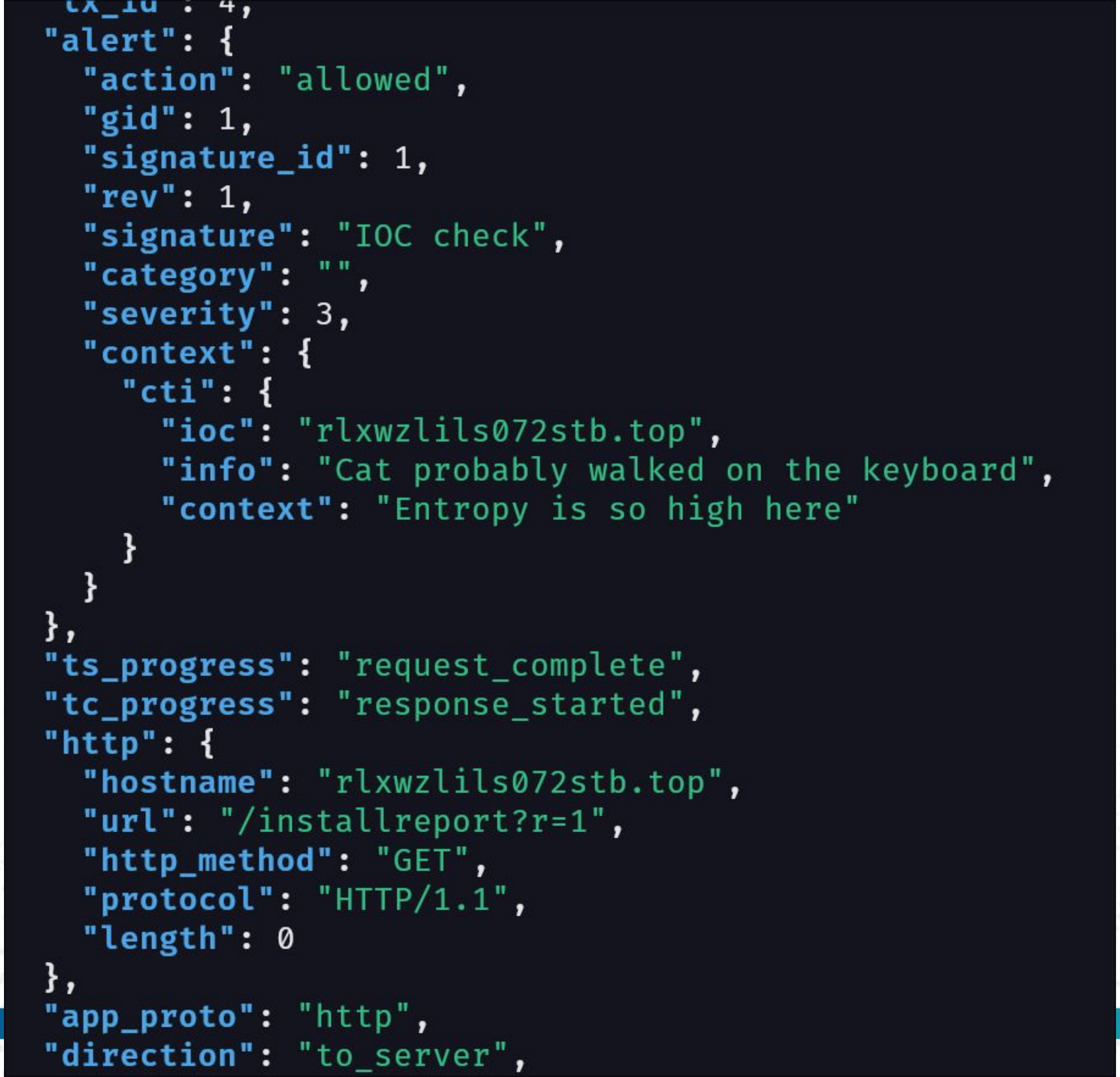
IOCs matching with dataset

- Don’t use one signature per IOC
- Better technique
  - Match on a list
  - With dataset keyword
- List entries in a file
  - In base64 if a string
- Use dataset keyword
  - Example:

```
tls.sni; domain; dataset:isset,domain_ioc,load dom_ioc.lst, type string
```

Demo: IOC matching with dataset with JSON

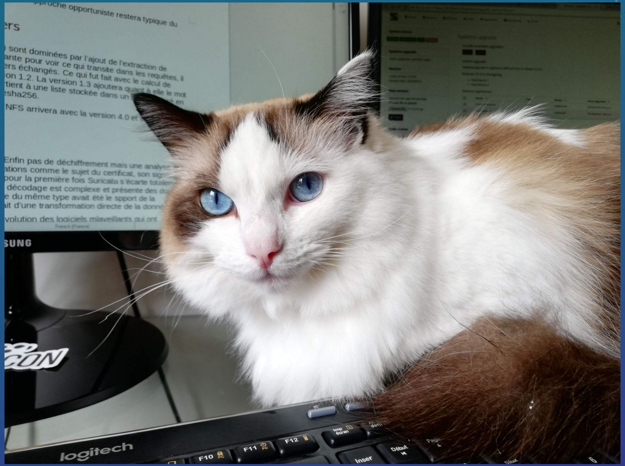
- Dataset with JSON
  - Dataset but information are attached to value
  - Needs Suricata 8.0.0
- Example:
  - tls.sni; dataset:isset,ioc,type string,load ioc.ndjson context_key circl, value_key host, format ndjson
  - In ioc.ndjson:
    - {“host”: “toto.com”, “type”: “plumber”, “country”: “japan”}
    - {“host”: “hack.lu”, “type”: “conference”, “country”: “Luxembourg”}
  - Match on “toto.com” produces:
    - … “alert”: {“context“: { “circl”: {“host”: “toto.com”, “type”: “plumber”, “country”: “japan”}} …



```
  "tx_id": 4,
  "alert": {
    "action": "allowed",
    "gid": 1,
    "signature_id": 1,
    "rev": 1,
    "signature": "IOC check",
    "category": "",
    "severity": 3,
    "context": {
      "cti": {
        "ioc": "rlxwzlils072stb.top",
        "info": "Cat probably walked on the keyboard",
        "context": "Entropy is so high here"
      }
    }
  },
  "ts_progress": "request_complete",
  "tc_progress": "response_started",
  "http": {
    "hostname": "rlxwzlils072stb.top",
    "url": "/installreport?r=1",
    "http_method": "GET",
    "protocol": "HTTP/1.1",
    "length": 0
  },
  "app_proto": "http",
  "direction": "to_server",
```

Suricata Language Server

By a lazy one, for the lazy ones



Challenge of writing signatures

- How do you know which keyword to use ?
- Is the syntax correct ?
- Did I wrote a signature with obvious performance issue ?

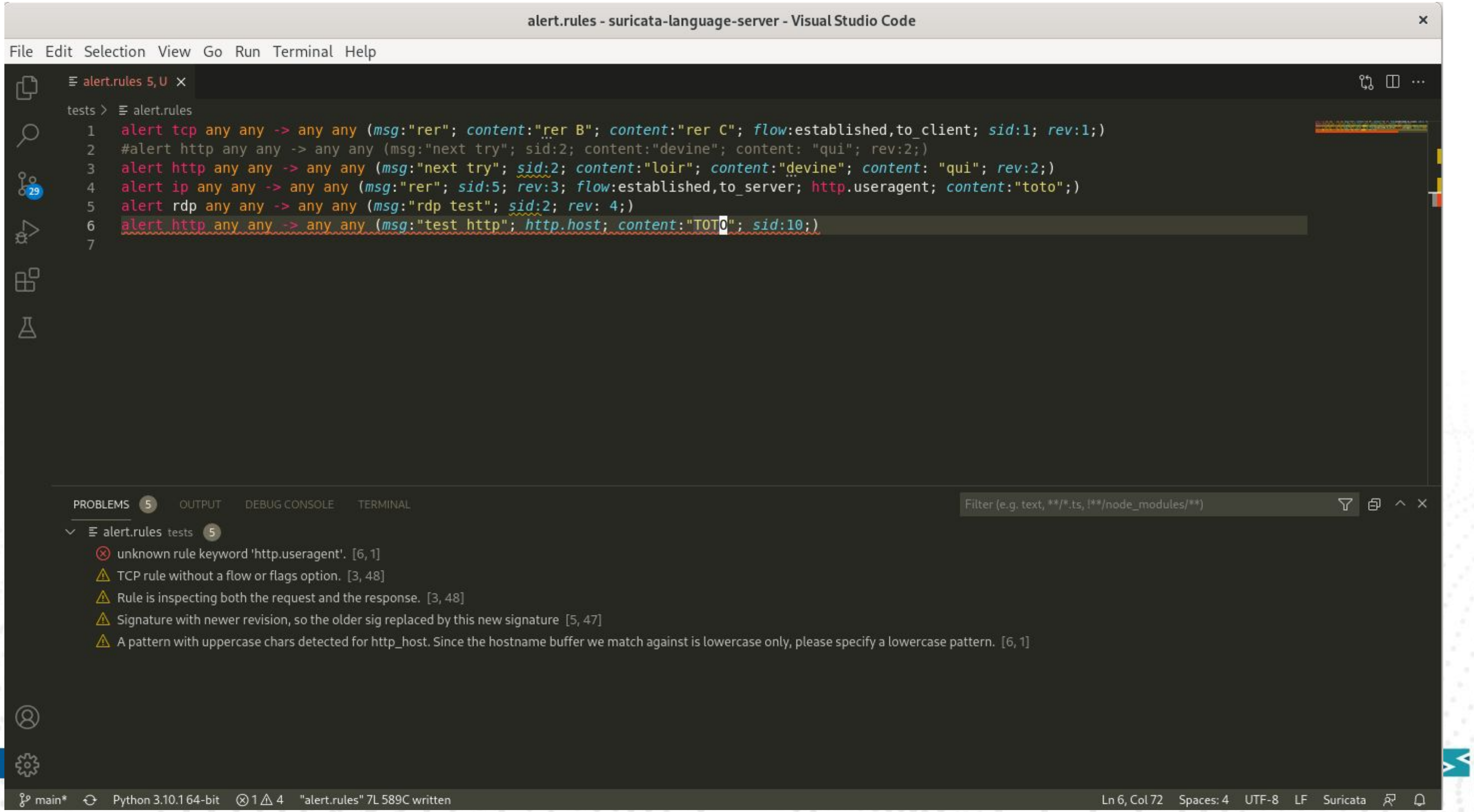
Excuse my French ? Language Server ?

- Language Server Protocol
  - Standardize communication between IDE/Text Editor and Language Server
  - Via JSON RPC
- Features
  - Auto complete
  - Go to definition
  - Find all references
  - Warnings
- More information:
  - https://microsoft.github.io/language-server-protocol/

Suricata Language Server

- A language server for Suricata signature
- Available under GPLv3 license
- Will get you
  - Syntax checking
  - Performance hints
    - Rule optimization
    - Detection engine optimization (fast pattern)
  - Auto completion
- In your preferred editors
- Tested with:
  - Visual Studio Code
  - Neovim
  - Kate
  - Sublime Text 3

VS Code

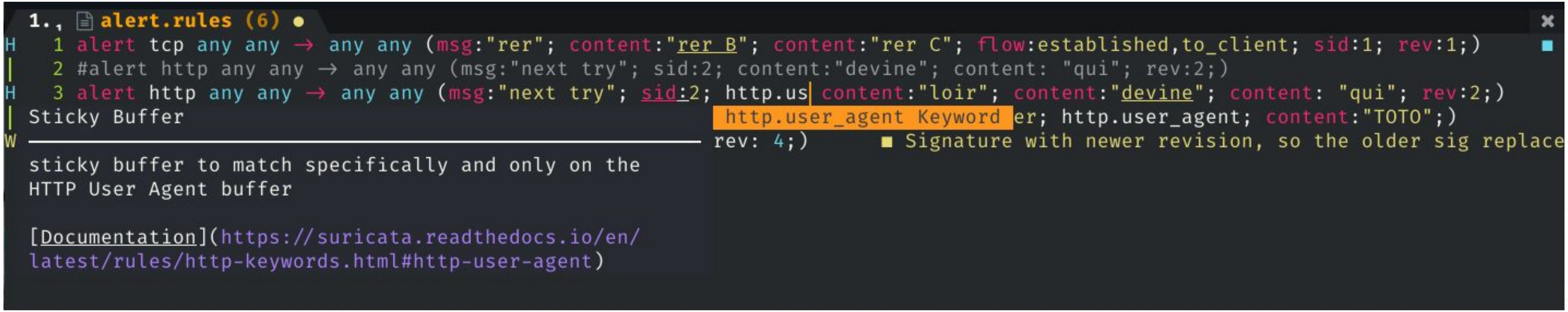


alert.rules - suricata-language-server - Visual Studio Code
File Edit Selection View Go Run Terminal Help
alert.rules 5, U
tests > alert.rules

```
1  alert tcp any any -> any any (msg:"rer"; content:"rer B"; content:"rer C"; flow:established,to_client; sid:1; rev:1;)
2  #alert http any any -> any any (msg:"next try"; sid:2; content:"devine"; content: "qui"; rev:2;)
3  alert http any any -> any any (msg:"next try"; sid:2; content:"loir"; content:"devine"; content: "qui"; rev:2;)
4  alert ip any any -> any any (msg:"rer"; sid:5; rev:3; flow:established,to_server; http.useragent; content:"toto";)
5  alert rdp any any -> any any (msg:"rdp test"; sid:2; rev: 4;)
6  alert http any any -> any any (msg:"test http"; http.host; content:"TOTO"; sid:10;)
7
```

- unknown rule keyword 'http.useragent'. [6, 1]
- TCP rule without a flow or flags option. [3, 48]
- Rule is inspecting both the request and the response. [3, 48]
- Signature with newer revision, so the older sig replaced by this new signature [5, 47]
- A pattern with uppercase chars detected for http_host. Since the hostname buffer we match against is lowercase only, please specify a lowercase pattern. [6, 1]

Neovim



```
 1., alert.rules (6) ●                                                                                                        ✖
H  alert tcp any any → any any (msg:"rer"; content:"rer B"; content:"rer C"; flow:established,to_client; sid:1; rev:1;)    ■
|  #alert http any any → any any (msg:"next try"; sid:2; content:"devine"; content: "qui"; rev:2;)
H  alert http any any → any any (msg:"next try"; sid:2; http.us content:"loir"; content:"devine"; content: "qui"; rev:2;)
| Sticky Buffer                                      http.user_agent Keyword er; http.user_agent; content:"TOTO";)
W ─────────────────────────────────────────────────  rev: 4;)      ■ Signature with newer revision, so the older sig replace
  sticky buffer to match specifically and only on the
  HTTP User Agent buffer

  [Documentation](https://suricata.readthedocs.io/en/
  latest/rules/http-keywords.html#http-user-agent)
```

SLS Installation

Time for fun

Hands on: Install SLS

- Available via pip:
  - pip install suricata-language-server
- Follow github for setup:
  - https://github.com/StamusNetworks/suricata-language-server

Editor setup

- See github page again
  - https://github.com/StamusNetworks/suricata-language-server
- Code & VSCode need also:
  - The SLS module
  - And the server

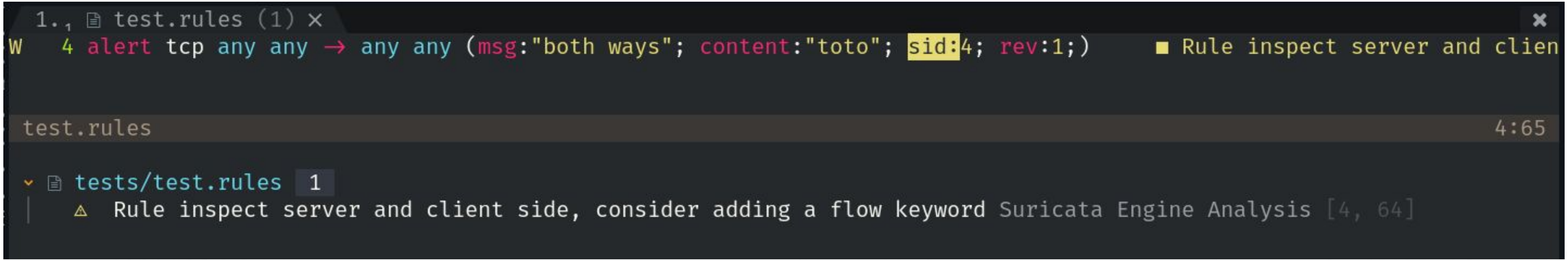
Making sense of warnings

Crash course on signature writing

STAMVS NETWORKS

Directionality warning

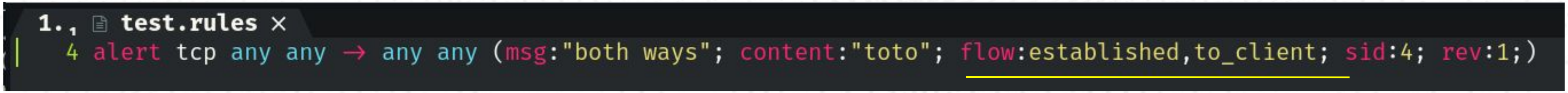
- Bad rule



```
alert tcp any any → any any (msg:"both ways"; content:"toto"; sid:4; rev:1;)     ■ Rule inspect server and clien
```

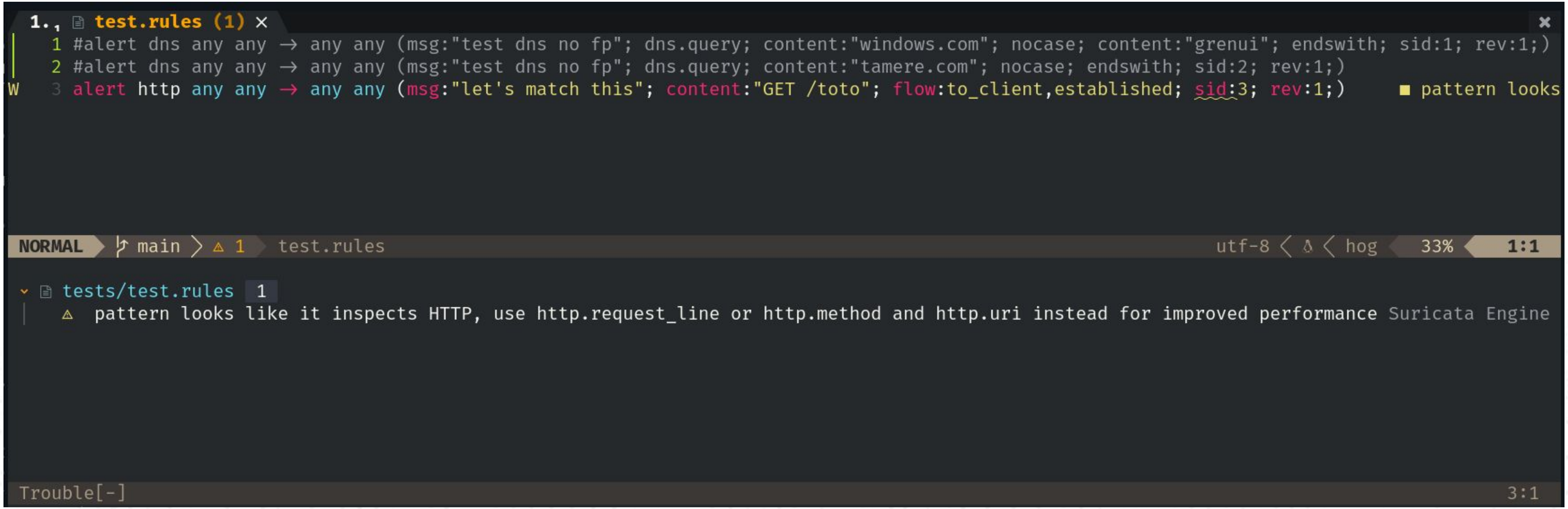
```
⚠ Rule inspect server and client side, consider adding a flow keyword Suricata Engine Analysis [4, 64]
```

- Correct one



```
alert tcp any any → any any (msg:"both ways"; content:"toto"; flow:established,to_client; sid:4; rev:1;)
```

Missing HTTP keywords



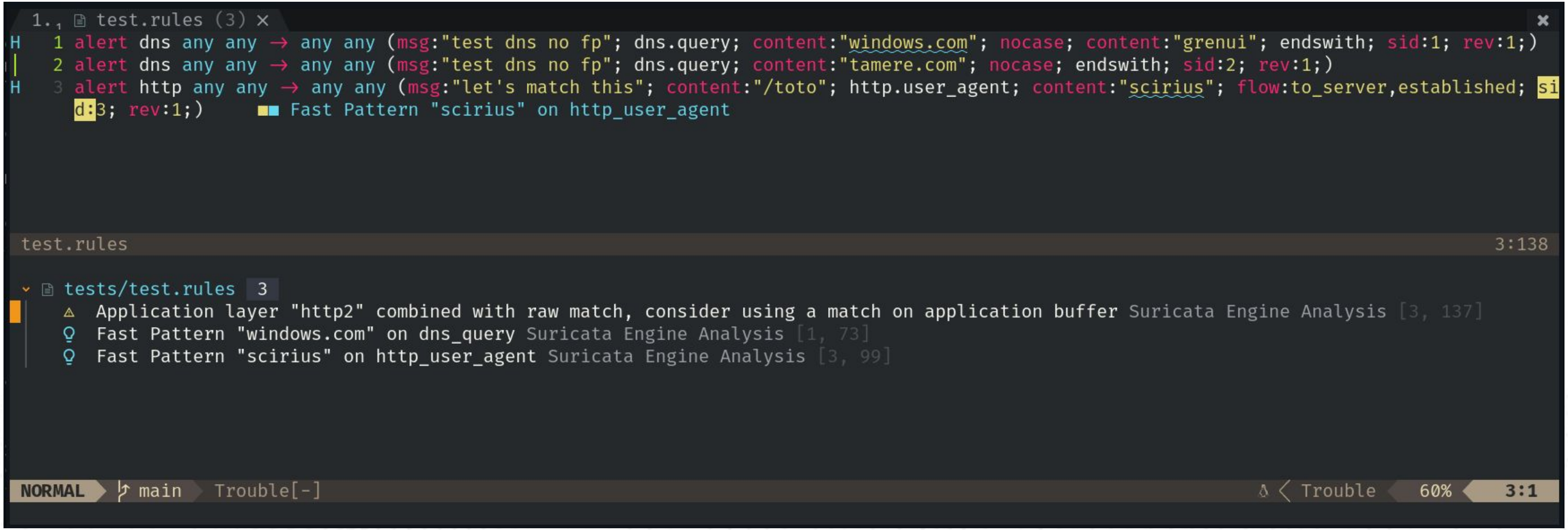
```
 1., 🗎 test.rules (1) ×                                                                                                ✖
 #alert dns any any → any any (msg:"test dns no fp"; dns.query; content:"windows.com"; nocase; content:"grenui"; endswith; sid:1; rev:1;)
 #alert dns any any → any any (msg:"test dns no fp"; dns.query; content:"tamere.com"; nocase; endswith; sid:2; rev:1;)
W alert http any any → any any (msg:"let's match this"; content:"GET /toto"; flow:to_client,established; sid:3; rev:1;)     ■ pattern looks

 NORMAL  main  ⚠ 1  test.rules                                                                     utf-8   hog  33%  1:1

 ⌄ 🗎 tests/test.rules  1
   ⚠  pattern looks like it inspects HTTP, use http.request_line or http.method and http.uri instead for improved performance Suricata Engine

 Trouble[-]                                                                                                                3:1
```

Mixed content



```
1., test.rules (3) ×
H   1 alert dns any any → any any (msg:"test dns no fp"; dns.query; content:"windows.com"; nocase; content:"grenui"; endswith; sid:1; rev:1;)
    2 alert dns any any → any any (msg:"test dns no fp"; dns.query; content:"tamere.com"; nocase; endswith; sid:2; rev:1;)
H   3 alert http any any → any any (msg:"let's match this"; content:"/toto"; http.user_agent; content:"scirius"; flow:to_server,established; si
      d:3; rev:1;)     ■■ Fast Pattern "scirius" on http_user_agent

test.rules                                                                                                                             3:138

⌄ tests/test.rules  3
     ⚠ Application layer "http2" combined with raw match, consider using a match on application buffer Suricata Engine Analysis [3, 137]
     Fast Pattern "windows.com" on dns_query Suricata Engine Analysis [1, 73]
     Fast Pattern "scirius" on http_user_agent Suricata Engine Analysis [3, 99]

NORMAL   main   Trouble[-]                                                                                      Trouble   60%   3:1
```

Hands on: write a signature

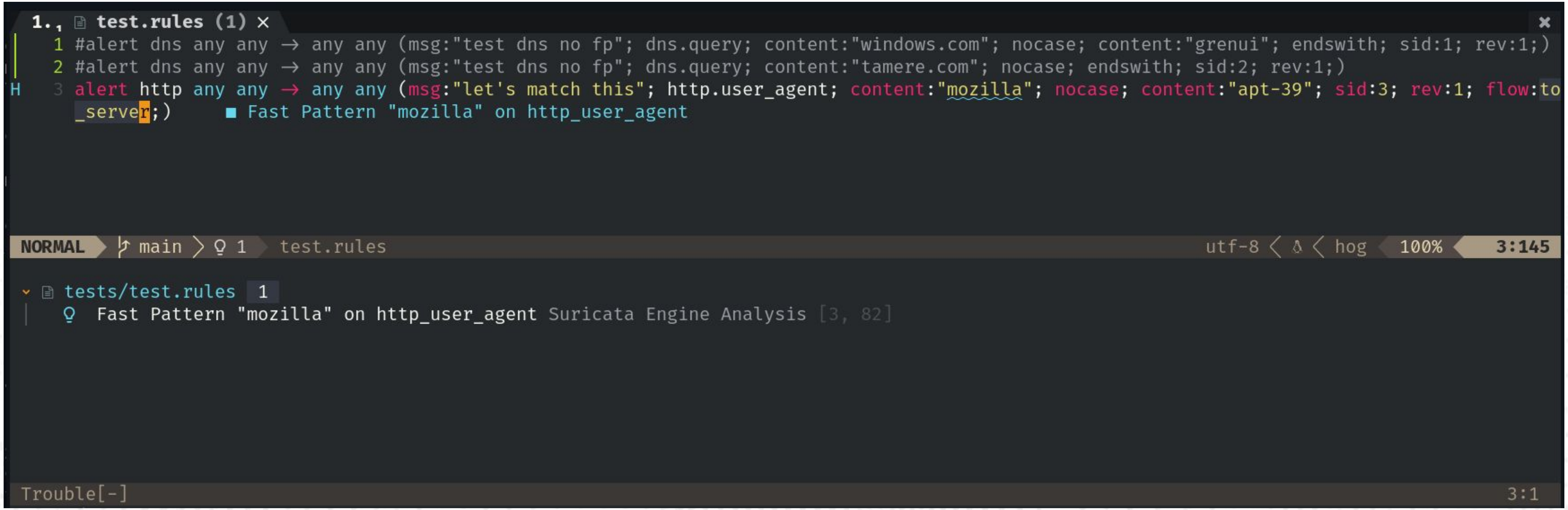
- It must:
  - Detect that the user agent follow pattern “Mozilla.*WindowsPowerShell.*4768”
  - Check that hostname is rlxwzlils072stb[.]top
- Success criteria
  - No warning in SLS
  - Alert one time only per source IP in 30 seconds
    - Using threshold keyword
- Why is it a bad idea to threshold in this example
  - Study protocol information in the alert without threshold

Fast pattern crash course

- Linear evaluation of ruleset is impossible
  - ETPro ruleset as more than 60000 rules
  - At 40Gbps this means 0.05 ns per rule
- Suricata needs to use heavy optimization
- Main one is multi pattern matching
  - Match multiple pattern in various buffer
  - Only evaluate signatures that contains the pattern
  - Ruleset MUST minimized the number of signature using same pattern
- How Suricata picks fast pattern ?
  - Longer pattern in content match
  - Follow *fast_pattern* keyword instruction

Fast pattern check

- Fast pattern is main optimization in detection engine
- Need to be done on a differentiator
- Suricata picks the longest pattern



```
1., test.rules (1) ×                                                                                    ✖
#alert dns any any -> any any (msg:"test dns no fp"; dns.query; content:"windows.com"; nocase; content:"grenui"; endswith; sid:1; rev:1;)
#alert dns any any -> any any (msg:"test dns no fp"; dns.query; content:"tamere.com"; nocase; endswith; sid:2; rev:1;)
alert http any any -> any any (msg:"let's match this"; http.user_agent; content:"mozilla"; nocase; content:"apt-39"; sid:3; rev:1; flow:to
_server;)      ■ Fast Pattern "mozilla" on http_user_agent

NORMAL  main  1  test.rules                                                      utf-8   hog  100%  3:145

tests/test.rules  1
  Fast Pattern "mozilla" on http_user_agent Suricata Engine Analysis [3, 82]

Trouble[-]                                                                                           3:1
```

Adapt to custom configurations

Rules edition is not done on probe

Problem:

- Signature validity depends on configuration
- Configuration is not known at edit time
- Examples:
  - Use of variables
  - Configuration not active by default
  - Depends on path not present on rules writer system

Solution:

- Method:
  - Use the rules file to do modify settings
- How:
  - Like editor does
  - Add comments to tune the analysis

Custom Suricata option

Syntax:

```
## SLS suricata-options: --set vars.address-groups.GATEWAY_SERVERS="127.0.0.3"
```

Useful to write signature with

- Custom variables
- Non active settings

Dataset handling

- Handling dataset
  - File may not be in final destination
  - Can be fixed with setup in most cases
- Syntax and example:

```
## SLS dataset-dir: /path/to/dataset/


alert http any any -> any any (msg:"test"; file_data; dataset:isset,fi,type string, load
/path/to/dataset/dd.lst; sid:1;)
```

Building a CI for Suricata rules

STAMVS NETWORKS

2 Objectives

- Linter
  - Verify syntax
  - Verify internal rules
- Performance
  - Check for impact of signatures on performance

Proposed Strategy

- SLS as a linter
    - Using batch-file option
    - Without engine-analysis
        - Syntax check
    - With engine-analysis
        - Can implement syntax check
- Suricata performance analysis
    - Replay time need reproducible condition
        - Hardware only
        - No shared resources
    - Suricata has a profiling mode for signature
        - Run big enough pcap
        - Compare output to detect abnormal CPU usage by signature

SLS batch mode

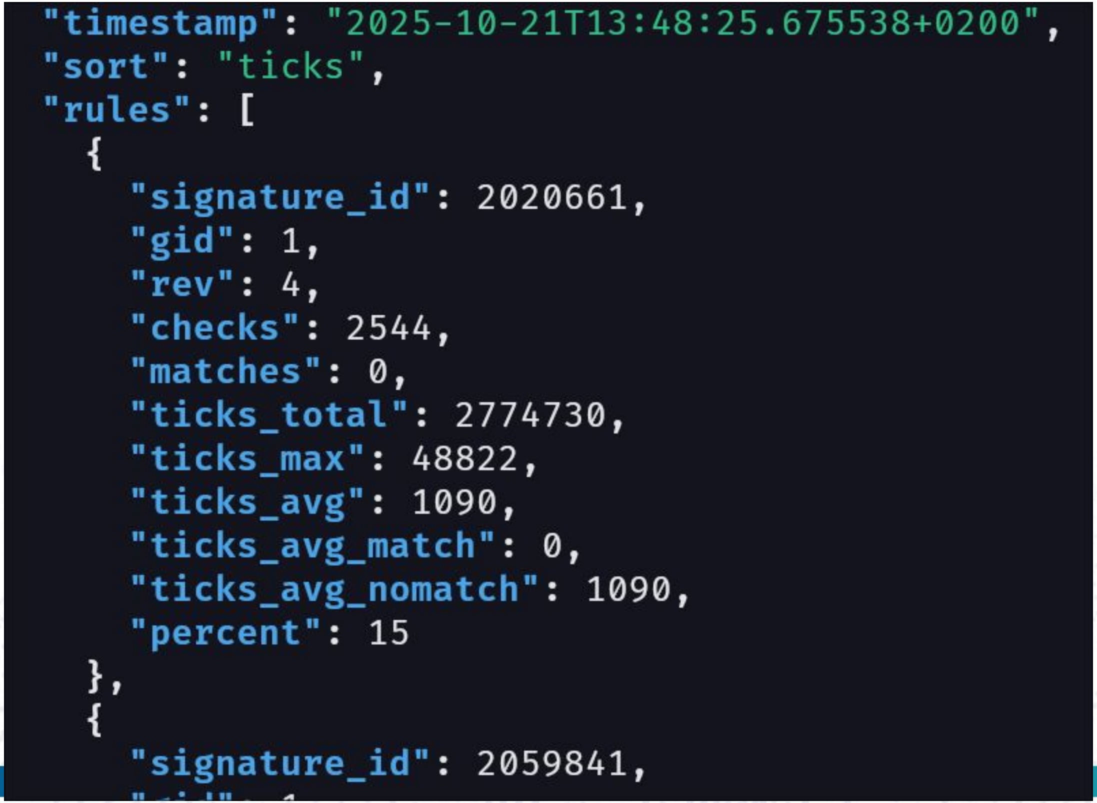
```
suricata-language-server --batch-file invalid-http-host.rules --suricata-binary /home/eric/builds/suricata/bin/suricata | jq 'select(.severity<4)'
```

Performance measurement

Suricata profiling mode

- Compute various CPU usage time
    - Dedicated mode for rules profiling
    - Or more global
- Get the number of ticks on rule
- Build option:
    - Need to be built in the binary
        - Use --enable-profiling
    - Really costly in performance
        - Better use --enable-rules-profiling

Understanding output format



```
"timestamp": "2025-10-21T13:48:25.675538+0200",
"sort": "ticks",
"rules": [
  {
    "signature_id": 2020661,
    "gid": 1,
    "rev": 4,
    "checks": 2544,
    "matches": 0,
    "ticks_total": 2774730,
    "ticks_max": 48822,
    "ticks_avg": 1090,
    "ticks_avg_match": 0,
    "ticks_avg_nomatch": 1090,
    "percent": 15
  },
  {
    "signature_id": 2059841,
    "gid": 1
```

Live performance measurement

- Full time profiling is killing performance
- On demand:
  - Activate computing of profiling
  - Use unix socket
    - suricatasc -c ruleset-profile-start >/dev/null
    - sleep 10
    - suricatasc -c ruleset-profile >> log.json
    - suricatasc -c ruleset-profile-stop >/dev/null
- Sampling:
  - Only analyze a subset of packets
  - If signature is costly, it will show

Demo

Suricata performance dashboard in Clear NDR Enterprise

Conclusion

Let’s wrap it up.

STAMVS NETWORKS

SLS Future

- New features planned:
  - SID conflict detection
  - Docker mode:
    - Skip Suricata installation
    - Run Suricata inside Docker container
- Ecosystem feature:
  - CI file for gitlab
  - Github action

Questions ?

- 2nd Suricata workshop:
  - New advanced network detection with Suricata 8
  - This afternoon (Wed 22th)
    - 2:15pm to 4:15pm
    - Vianden & Wiltz
- Our talk:
  - What's new in Suricata 8, Thursday 23th, 5pm-5:30pm
- Links:
  - Suricata: https://suricata.io/
  - Suricata Language Server: https://www.stamus-networks.com/suricata-language-server
  - Suricata for Analysts: https://www.stamus-networks.com/suricata-4-analysts
  - Clear NDR Community: https://www.stamus-networks.com/clear-ndr-community