Suricata

What is an IDS and Network Security Monitoring in 2023?

Agenda

About us

What is Suricata

How it started

How it evolved

Challenges when monitoring traffic

How to get involved/contribute and stay in touch

Eric Leblond

CTO at Stamus Networks
OISF Team - Developer/Trainer
OISF Board of Directors
Linux Kernel/Netfilter developer
Scirius CE/SELKS maintainer
@regit @regiteric

Peter Manev

@pevma

13 yrs with Suricata

OISF Exec team

Suricata QA/Training lead

CSO Stamus Networks

SELKS maintainer

Me likes -

Open Source

Threat Hunting

What is Suricata

What is Suricata?

- A high-performance network monitoring and security engine with active/passive monitoring, metadata logging and real-time file identification and extraction

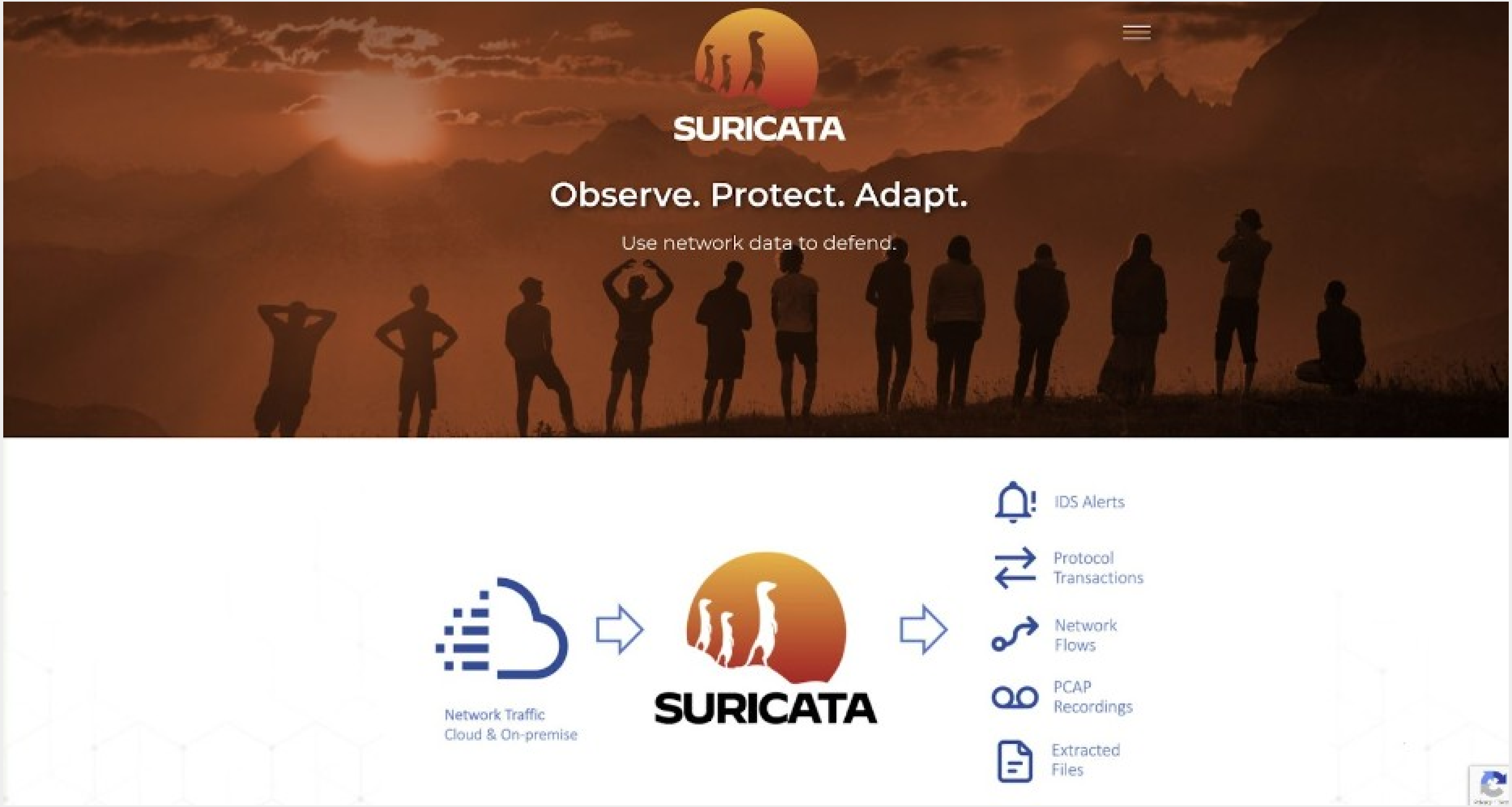
- Powered by Open Source GPLv2 - find it on Github:
  - https://github.com/OISF/suricata
- Produces a high-level of situational awareness and detailed application layer transaction records from network traffic.

- Used by thousands of organisations and ppl around the globe

What is Suricata ?

Suricata can be deployed as

- **IDS** - Intrusion Detection System (passive sniffing)
- **IPS** - Intrusion Prevention system (inline)
- **NSM** - Network Security Monitoring (works without rules)
  - Protocol , flow and filetranscation logging
- **FPC** - Full Pcap Capture
  - Also possible: **Conditional** PCAP Capture
    - Thanks Eric Leblond **!**
- Combinations of the above like
  - IDS + NSM + FPC
  - IDS + Conditional PCAP capture



SURICATA

Observe. Protect. Adapt.

Use network data to defend.

Suricata - Major Features

- Standards based formats (YAML, JSON) ease integrations with SIEM tools such as Elastic and Splunk
- Multithreaded, hardware acceleration available. 100Gb+ deployments
- Network metadata logging for a variety of protocols
- Advanced HTTP, DNS, SMTP, SMB and TLS support
- File identification and extraction - FTP/SMTP/HTTP/HTTP2/NFS/SMBv1-3
- Support for SCADA protocols - DNP3, ENIP, and CIP

Why The Network?

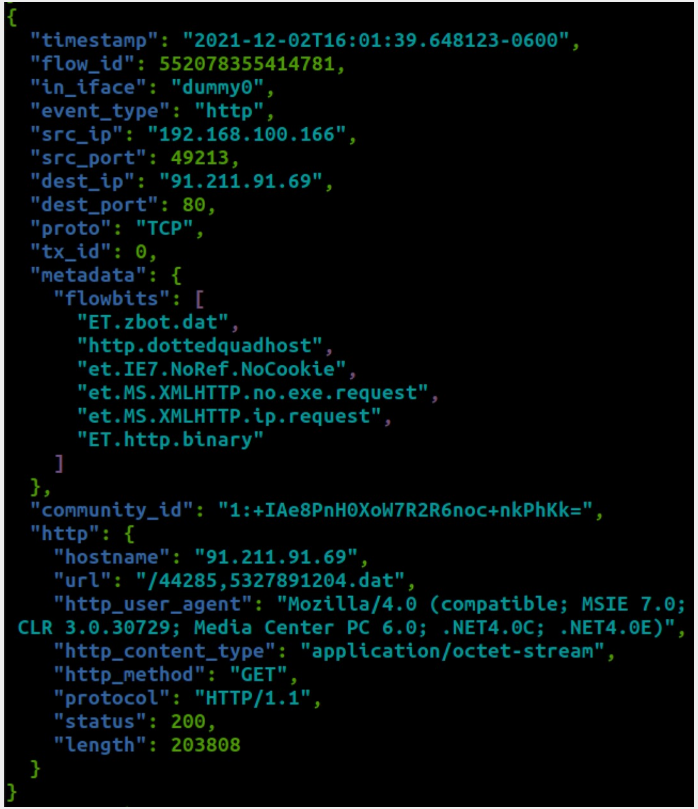
- The network is now the backbone of society
  - Connects computers for everything from social media to finance
- Criminals and other threat actors also utilize the network:
  - To attack the user
  - To deliver malware and other tools
  - To steal data
- Monitoring the network helps you to identify and stop this malicious activity

Network Metadata Logging

- Provides extensive logging of protocol and other network data

- Data logged in event records: HTTP/HTTP2, DNS, FTP, TLS, SMB, SSH, RDP…

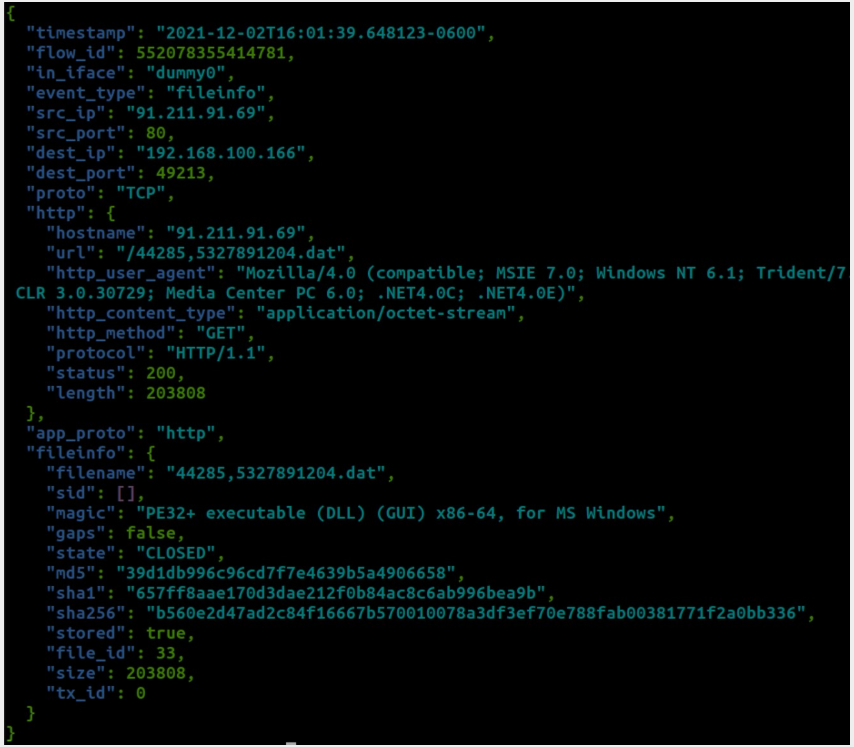
- Default output format in **J**ava**S**cript **O**bject **N**otation (JSON)



```
{
  "timestamp": "2021-12-02T16:01:39.648123-0600",
  "flow_id": 552078355414781,
  "in_iface": "dummy0",
  "event_type": "http",
  "src_ip": "192.168.100.166",
  "src_port": 49213,
  "dest_ip": "91.211.91.69",
  "dest_port": 80,
  "proto": "TCP",
  "tx_id": 0,
  "metadata": {
    "flowbits": [
      "ET.zbot.dat",
      "http.dottedquadhost",
      "et.IE7.NoRef.NoCookie",
      "et.MS.XMLHTTP.no.exe.request",
      "et.MS.XMLHTTP.ip.request",
      "ET.http.binary"
    ]
  },
  "community_id": "1:+IAe8PnH0XoW7R2R6noc+nkPhKk=",
  "http": {
    "hostname": "91.211.91.69",
    "url": "/44285,5327891204.dat",
    "http_user_agent": "Mozilla/4.0 (compatible; MSIE 7.0;
 CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; .NET4.0E)",
    "http_content_type": "application/octet-stream",
    "http_method": "GET",
    "protocol": "HTTP/1.1",
    "status": 200,
    "length": 203808
  }
}
```

File Identification and Extraction

- Can perform file identification and extraction in real-time
- File information includes:
  - Content type/libmagic
  - File hashes (MD5/SHA1/SHA2)
  - File size
- Files can also be extracted and stored to the file system

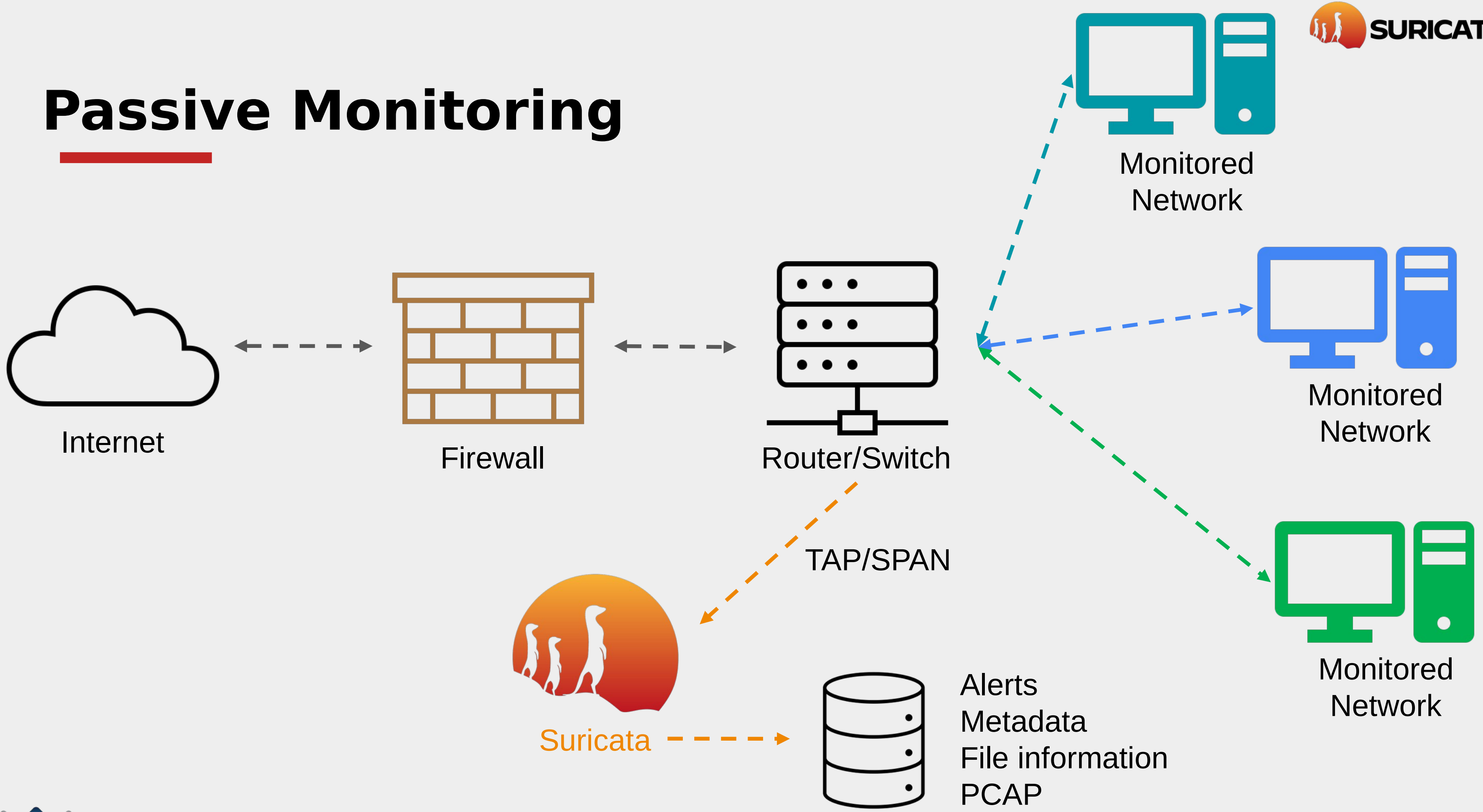


```
{
  "timestamp": "2021-12-02T16:01:39.648123-0600",
  "flow_id": 552078355414781,
  "in_iface": "dummy0",
  "event_type": "fileinfo",
  "src_ip": "91.211.91.69",
  "src_port": 80,
  "dest_ip": "192.168.100.166",
  "dest_port": 49213,
  "proto": "TCP",
  "http": {
    "hostname": "91.211.91.69",
    "url": "/44285,5327891204.dat",
    "http_user_agent": "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Trident/7
 CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; .NET4.0E)",
    "http_content_type": "application/octet-stream",
    "http_method": "GET",
    "protocol": "HTTP/1.1",
    "status": 200,
    "length": 203808
  },
  "app_proto": "http",
  "fileinfo": {
    "filename": "44285,5327891204.dat",
    "sid": [],
    "magic": "PE32+ executable (DLL) (GUI) x86-64, for MS Windows",
    "gaps": false,
    "state": "CLOSED",
    "md5": "39d1db996c96cd7f7e4639b5a4906658",
    "sha1": "657ff8aae170d3dae212f0b84ac8c6ab996bea9b",
    "sha256": "b560e2d47ad2c84f16667b570010078a3df3ef70e788fab00381771f2a0bb336",
    "stored": true,
    "file_id": 33,
    "size": 203808,
    "tx_id": 0
  }
}
```

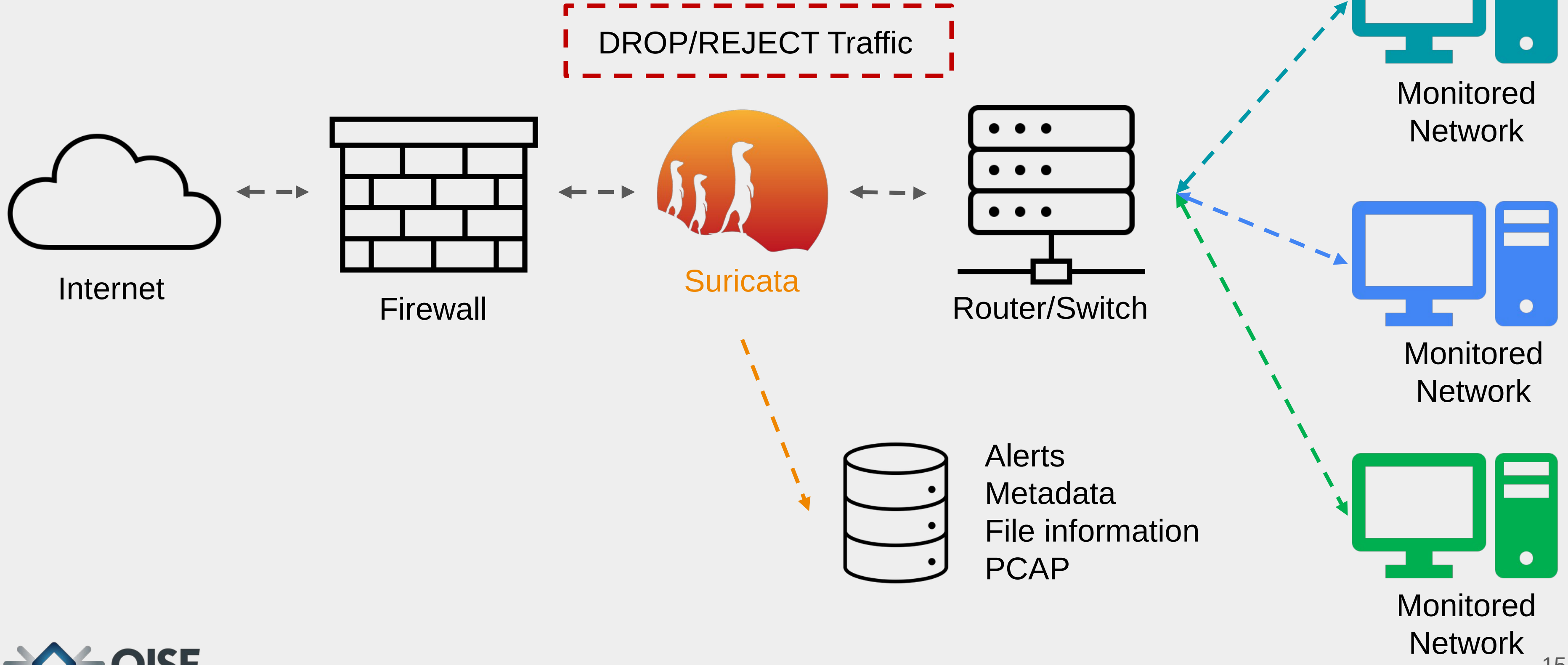
PCAP Capabilities

- Suricata can read PCAPs for offline processing
  - Ability to read a single PCAP or an entire directory
  - Can also process PCAPs through a Unix socket
- Suricata can also produce full packet capture (FPC)
  - Stored network data in PCAP files
- Consider multiple Suricata instances for testing/exploration/malware analysis

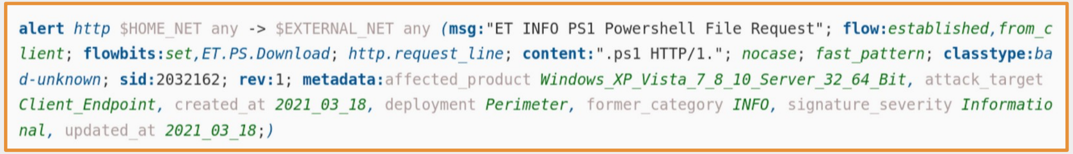
Passive Monitoring



Active Monitoring

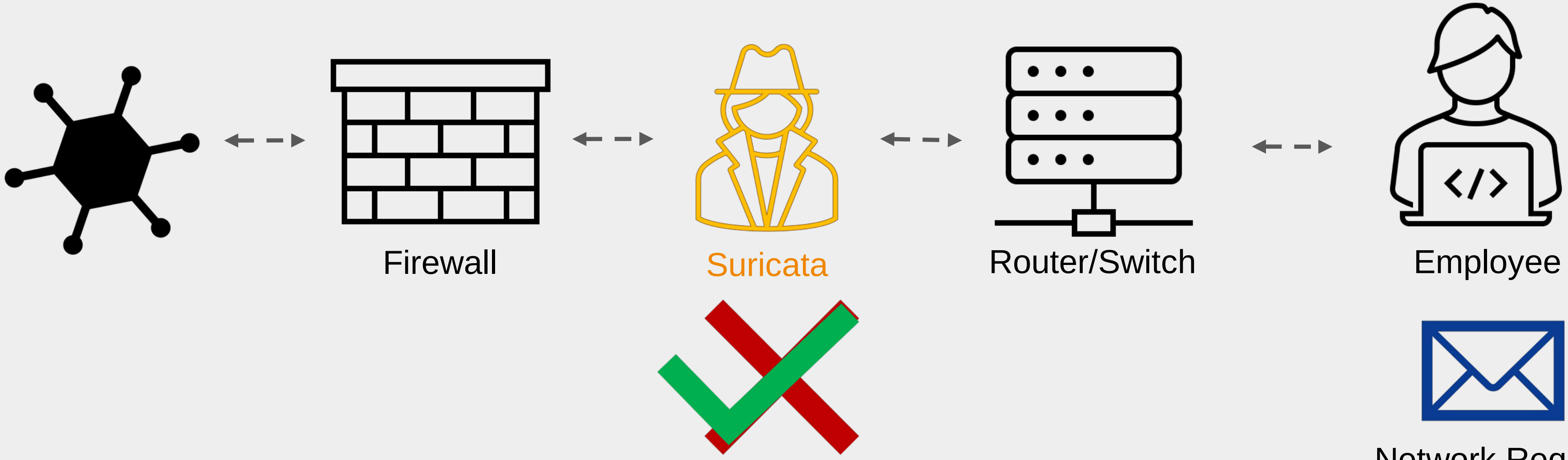


How Signatures Work



```
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ET INFO PS1 Powershell File Request"; flow:established,from_c
lient; flowbits:set,ET.PS.Download; http.request_line; content:".ps1 HTTP/1."; nocase; fast_pattern; classtype:ba
d-unknown; sid:2032162; rev:1; metadata:affected_product Windows_XP_Vista_7_8_10_Server_32_64_Bit, attack_target
Client_Endpoint, created_at 2021_03_18, deployment Perimeter, former_category INFO, signature_severity Informatio
nal, updated_at 2021_03_18;)
```

Malicious Document



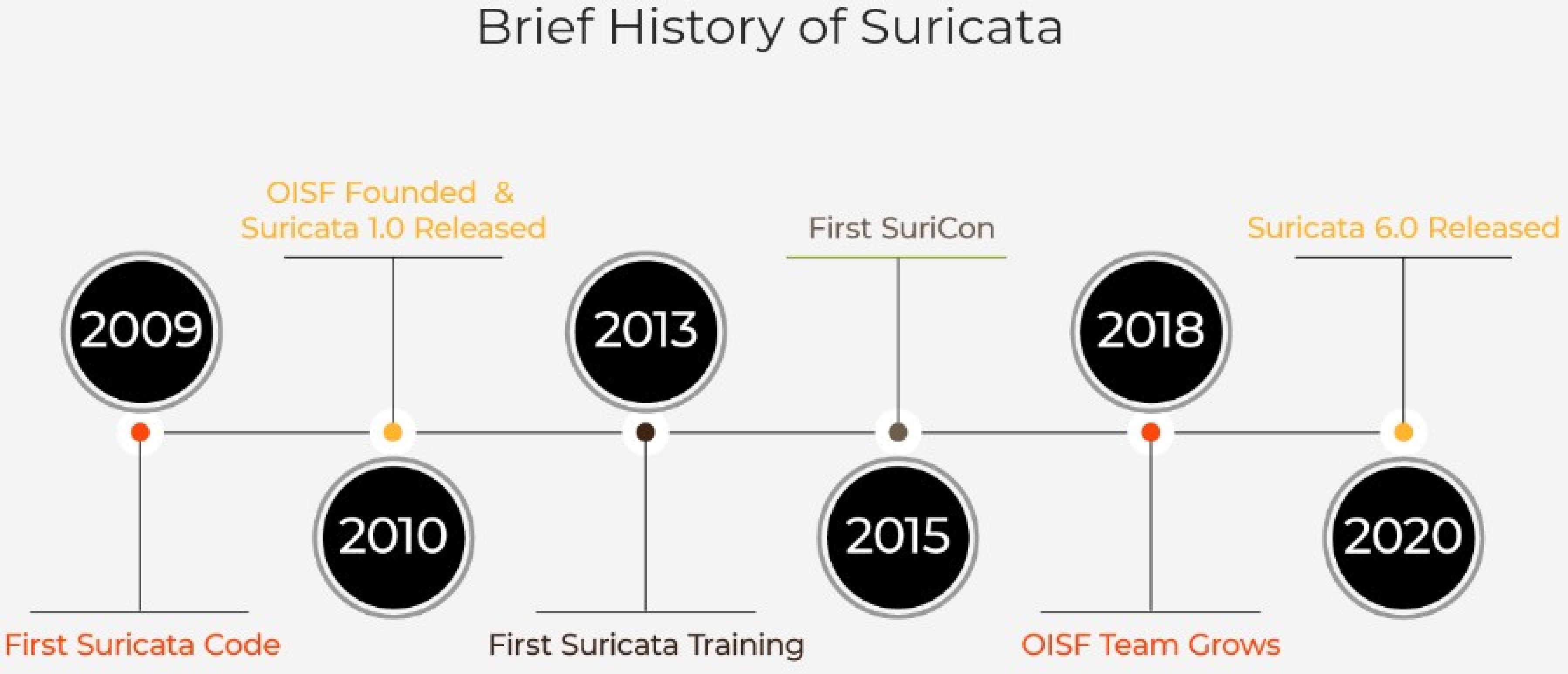
Network Request

Suricata History

Suricata History

- First lines of code written in 2007 by Victor Julien
  - First released in 2009
- Powered by Open Source GPLv2 (source on GitHub)
- Worked on/Developed with a global open source community in over 23 different countries
- Owned and supported by Open Information Security Foundation, a 501(c)3 non-profit
  - https://oisf.net

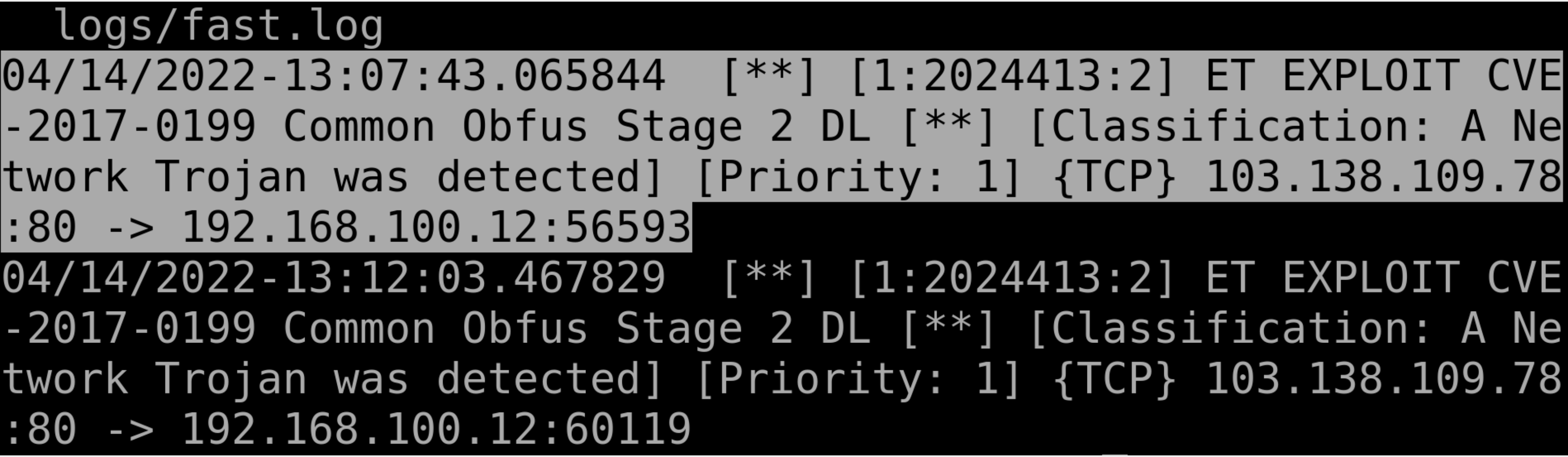
Suricata History



What is Suricata ?

How it started ?

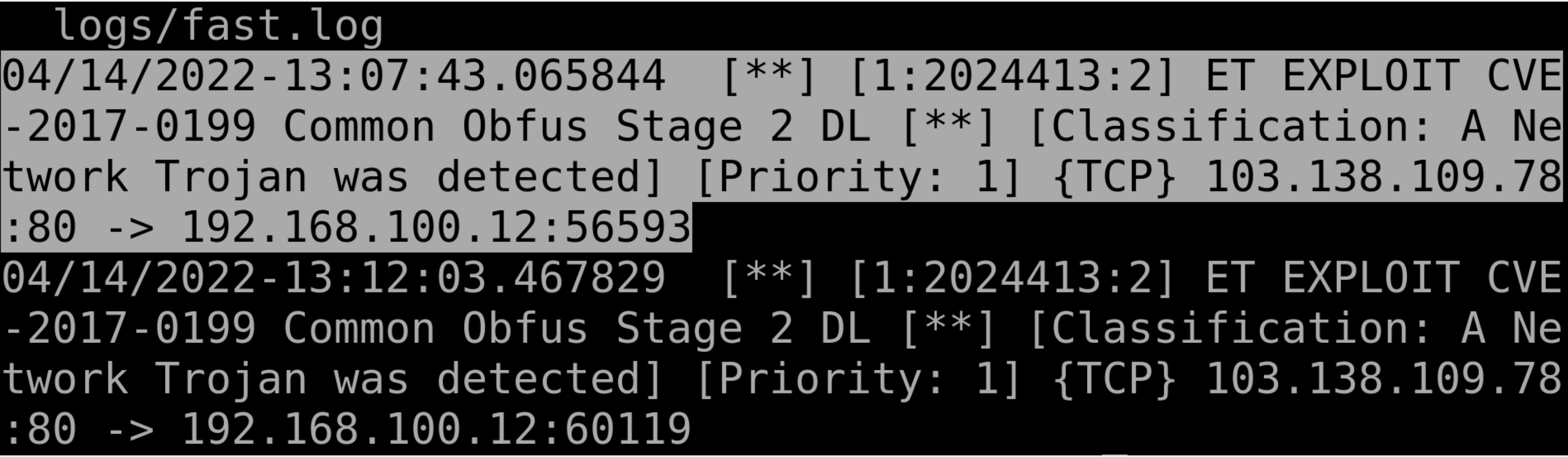
- An example of how **IDS** alert looked back **14+ yrs** ago



```
  logs/fast.log
04/14/2022-13:07:43.065844  [**] [1:2024413:2] ET EXPLOIT CVE
-2017-0199 Common Obfus Stage 2 DL [**] [Classification: A Ne
twork Trojan was detected] [Priority: 1] {TCP} 103.138.109.78
:80 -> 192.168.100.12:56593
04/14/2022-13:12:03.467829  [**] [1:2024413:2] ET EXPLOIT CVE
-2017-0199 Common Obfus Stage 2 DL [**] [Classification: A Ne
twork Trojan was detected] [Priority: 1] {TCP} 103.138.109.78
:80 -> 192.168.100.12:60119
```

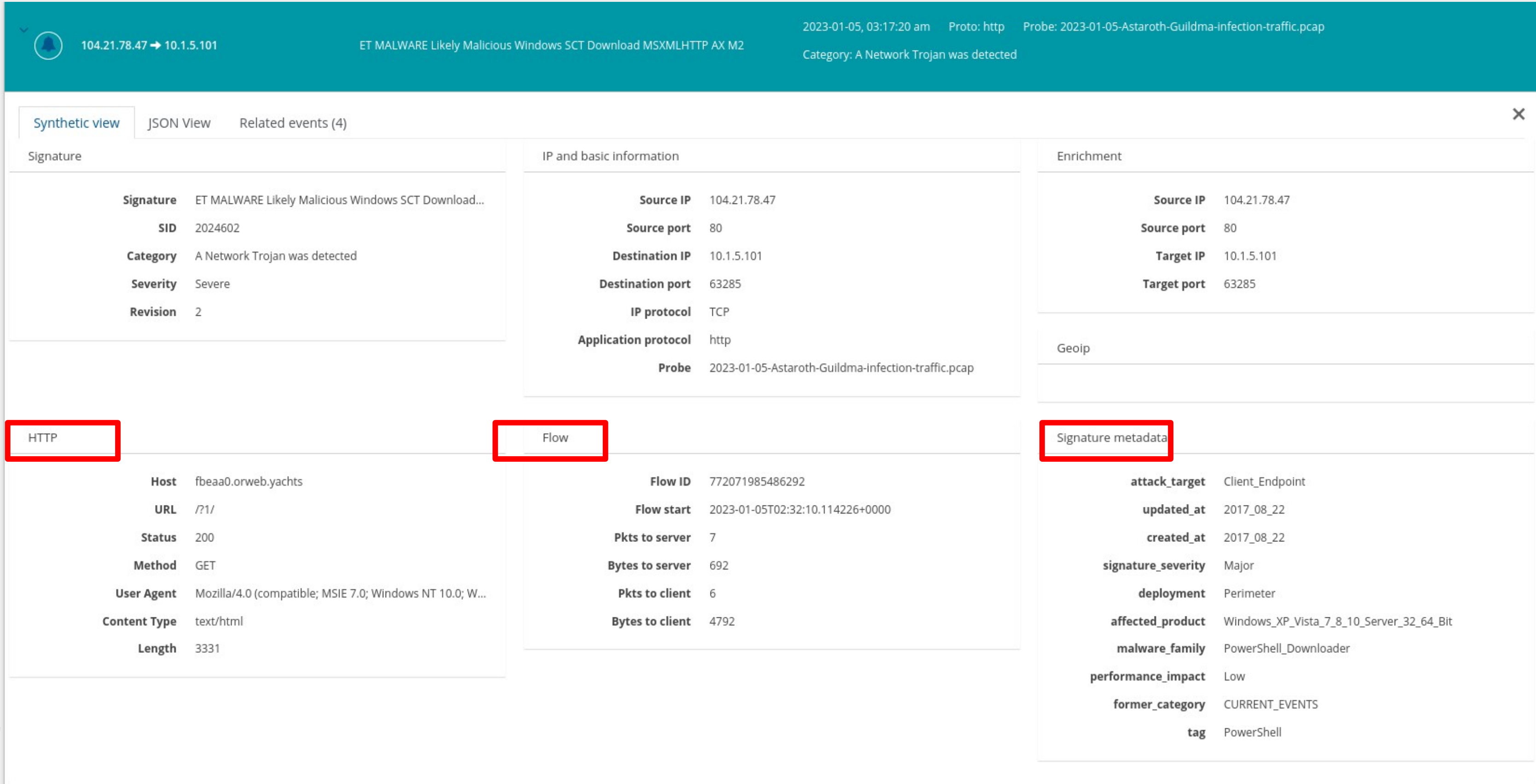
What is Suricata ?

- **14 yrs** ago - You had to go deploy other tools to find the logs related to this event and figure out if it is TP or FP



```
  logs/fast.log
04/14/2022-13:07:43.065844  [**] [1:2024413:2] ET EXPLOIT CVE
-2017-0199 Common Obfus Stage 2 DL [**] [Classification: A Ne
twork Trojan was detected] [Priority: 1] {TCP} 103.138.109.78
:80 -> 192.168.100.12:56593
04/14/2022-13:12:03.467829  [**] [1:2024413:2] ET EXPLOIT CVE
-2017-0199 Common Obfus Stage 2 DL [**] [Classification: A Ne
twork Trojan was detected] [Priority: 1] {TCP} 103.138.109.78
:80 -> 192.168.100.12:60119
```

What is Suricata ? How it looks today ?



104.21.78.47 → 10.1.5.101

ET MALWARE Likely Malicious Windows SCT Download MSXMLHTTP AX M2

2023-01-05, 03:17:20 am Proto: http Probe: 2023-01-05-Astaroth-Guildma-infection-traffic.pcap

Category: A Network Trojan was detected

Signature

Signature	ET MALWARE Likely Malicious Windows SCT Download...
SID	2024602
Category	A Network Trojan was detected
Severity	Severe
Revision	2

IP and basic information

Source IP	104.21.78.47
Source port	80
Destination IP	10.1.5.101
Destination port	63285
IP protocol	TCP
Application protocol	http
Probe	2023-01-05-Astaroth-Guildma-infection-traffic.pcap

Enrichment

Source IP	104.21.78.47
Source port	80
Target IP	10.1.5.101
Target port	63285

Geoip

HTTP

Host	fbeaa0.orweb.yachts
URL	/?1/
Status	200
Method	GET
User Agent	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 10.0; W...
Content Type	text/html
Length	3331

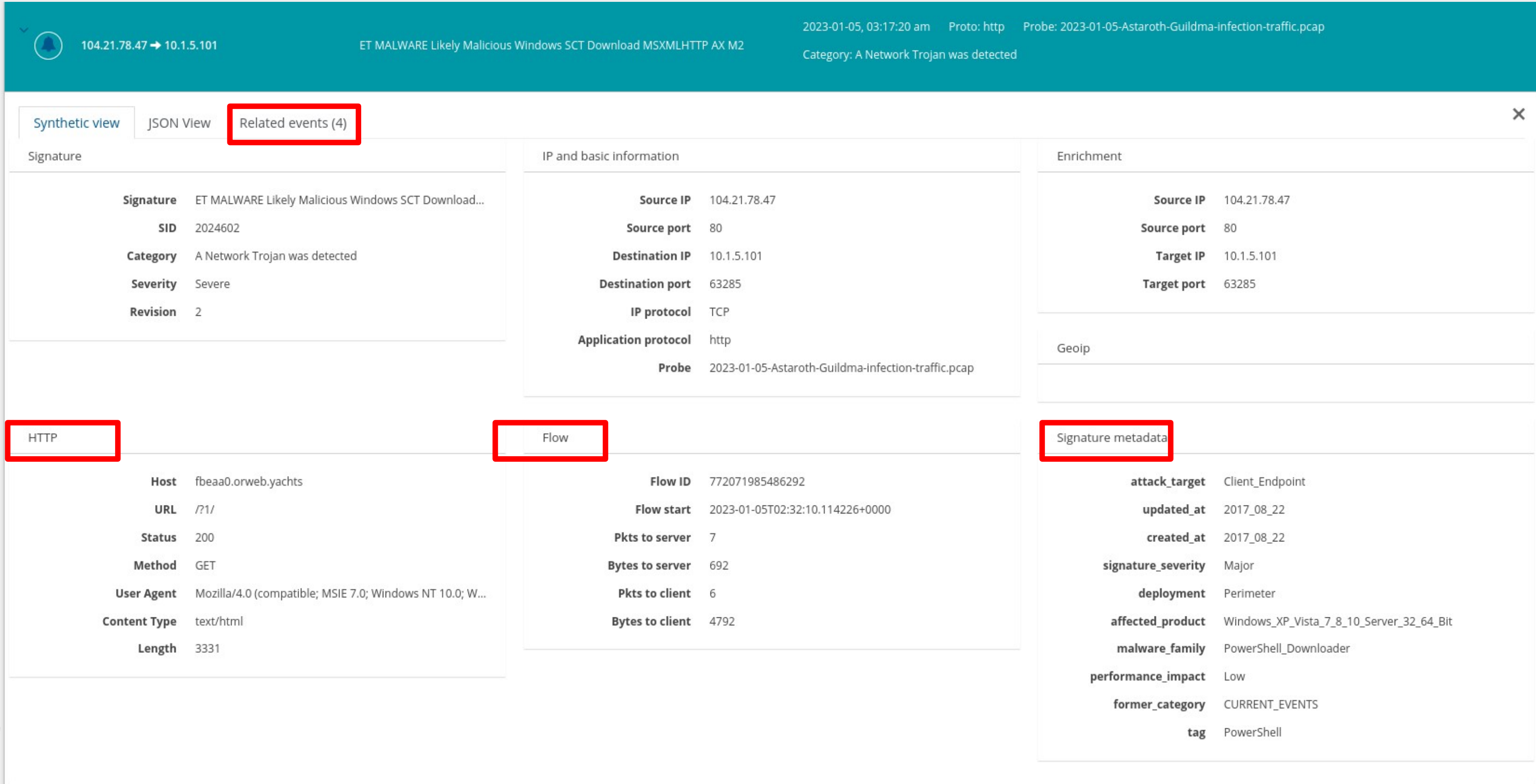
Flow

Flow ID	772071985486292
Flow start	2023-01-05T02:32:10.114226+0000
Pkts to server	7
Bytes to server	692
Pkts to client	6
Bytes to client	4792

Signature metadata

attack_target	Client_Endpoint
updated_at	2017_08_22
created_at	2017_08_22
signature_severity	Major
deployment	Perimeter
affected_product	Windows_XP_Vista_7_8_10_Server_32_64_Bit
malware_family	PowerShell_Downloader
performance_impact	Low
former_category	CURRENT_EVENTS
tag	PowerShell

What is Suricata ? How it looks today ?



104.21.78.47 ➔ 10.1.5.101

ET MALWARE Likely Malicious Windows SCT Download MSXMLHTTP AX M2

2023-01-05, 03:17:20 am Proto: http Probe: 2023-01-05-Astaroth-Guildma-infection-traffic.pcap

Category: A Network Trojan was detected

Signature

Signature	ET MALWARE Likely Malicious Windows SCT Download...
SID	2024602
Category	A Network Trojan was detected
Severity	Severe
Revision	2

IP and basic information

Source IP	104.21.78.47
Source port	80
Destination IP	10.1.5.101
Destination port	63285
IP protocol	TCP
Application protocol	http
Probe	2023-01-05-Astaroth-Guildma-infection-traffic.pcap

Enrichment

Source IP	104.21.78.47
Source port	80
Target IP	10.1.5.101
Target port	63285

Geoip

HTTP

Host	fbeaa0.orweb.yachts
URL	/?1/
Status	200
Method	GET
User Agent	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 10.0; W...
Content Type	text/html
Length	3331

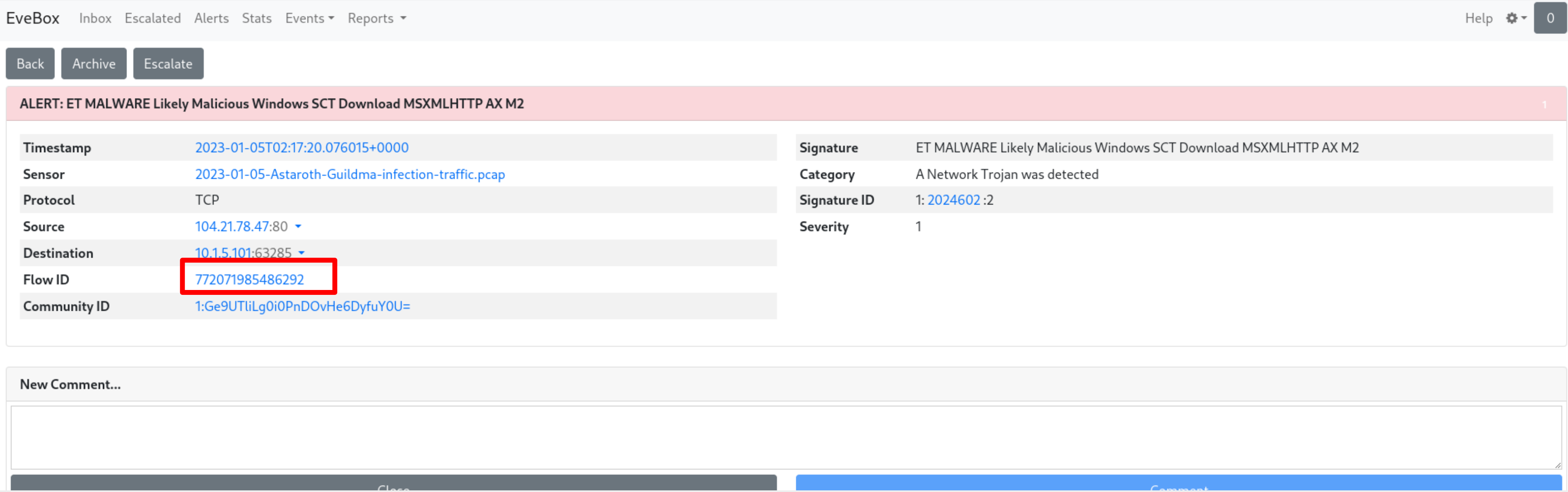
Flow

Flow ID	772071985486292
Flow start	2023-01-05T02:32:10.114226+0000
Pkts to server	7
Bytes to server	692
Pkts to client	6
Bytes to client	4792

Signature metadata

attack_target	Client_Endpoint
updated_at	2017_08_22
created_at	2017_08_22
signature_severity	Major
deployment	Perimeter
affected_product	Windows_XP_Vista_7_8_10_Server_32_64_Bit
malware_family	PowerShell_Downloader
performance_impact	Low
former_category	CURRENT_EVENTS
tag	PowerShell

What is Suricata ? How it looks today ?



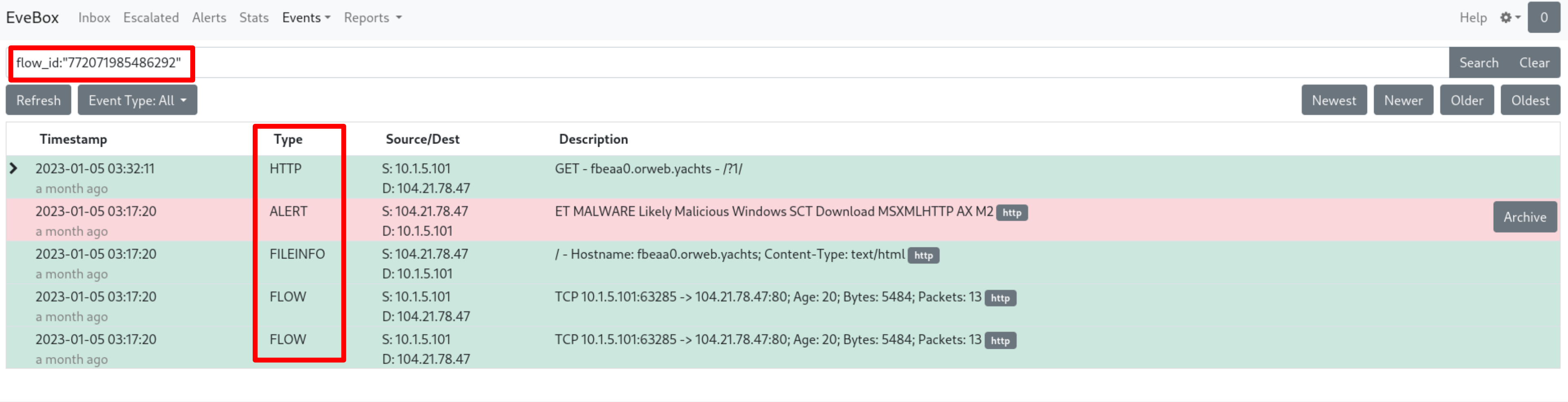
ALERT: ET MALWARE Likely Malicious Windows SCT Download MSXMLHTTP AX M2

Timestamp	2023-01-05T02:17:20.076015+0000
Sensor	2023-01-05-Astaroth-Guildma-infection-traffic.pcap
Protocol	TCP
Source	104.21.78.47:80
Destination	10.1.5.101:63285
Flow ID	772071985486292
Community ID	1:Ge9UTliLg0i0PnDOvHe6DyfuY0U=

Signature	ET MALWARE Likely Malicious Windows SCT Download MSXMLHTTP AX M2
Category	A Network Trojan was detected
Signature ID	1: 2024602 :2
Severity	1

EveBox - Showcasing Flow ID
https://evebox.org/

What is Suricata ? How it looks today ?

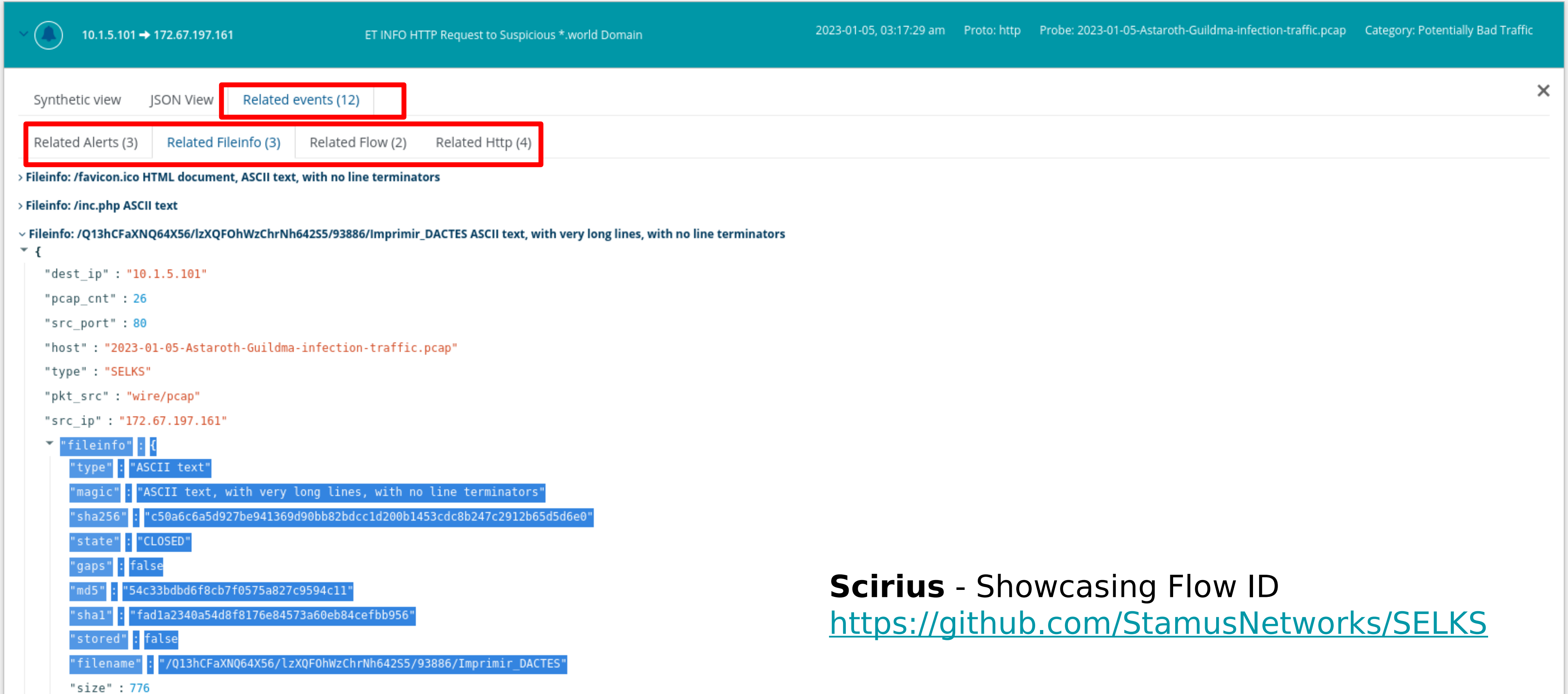


flow_id:"772071985486292"

Timestamp	Type	Source/Dest	Description
2023-01-05 03:32:11 a month ago	HTTP	S: 10.1.5.101 D: 104.21.78.47	GET - fbeaa0.orweb.yachts - /?1/
2023-01-05 03:17:20 a month ago	ALERT	S: 104.21.78.47 D: 10.1.5.101	ET MALWARE Likely Malicious Windows SCT Download MSXMLHTTP AX M2 http
2023-01-05 03:17:20 a month ago	FILEINFO	S: 104.21.78.47 D: 10.1.5.101	/ - Hostname: fbeaa0.orweb.yachts; Content-Type: text/html http
2023-01-05 03:17:20 a month ago	FLOW	S: 10.1.5.101 D: 104.21.78.47	TCP 10.1.5.101:63285 -> 104.21.78.47:80; Age: 20; Bytes: 5484; Packets: 13 http
2023-01-05 03:17:20 a month ago	FLOW	S: 10.1.5.101 D: 104.21.78.47	TCP 10.1.5.101:63285 -> 104.21.78.47:80; Age: 20; Bytes: 5484; Packets: 13 http

EveBox - Showcasing Flow ID
https://evebox.org/

What is Suricata ? How it looks today ?



› Fileinfo: /favicon.ico HTML document, ASCII text, with no line terminators

› Fileinfo: /inc.php ASCII text

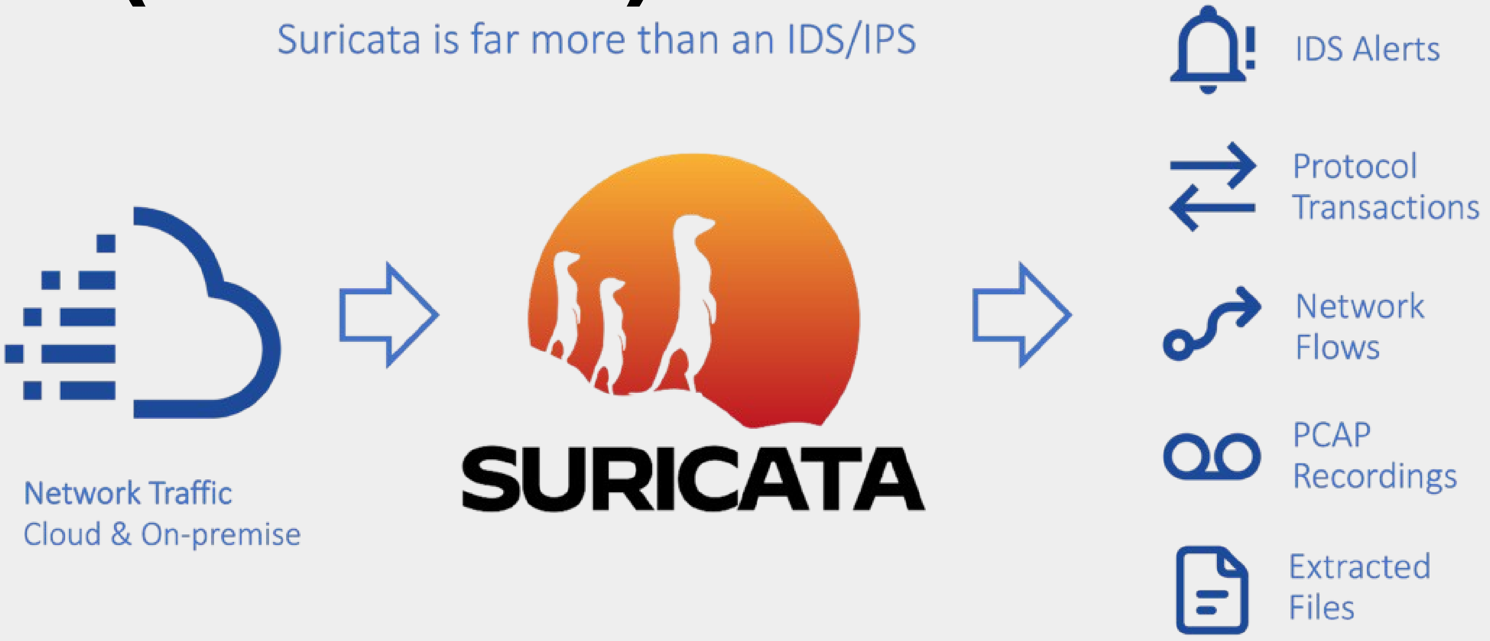
⌄ Fileinfo: /Q13hCFaXNQ64X56/lzXQFOhWzChrNh642S5/93886/Imprimir_DACTES ASCII text, with very long lines, with no line terminators

```
{
  "dest_ip" : "10.1.5.101"
  "pcap_cnt" : 26
  "src_port" : 80
  "host" : "2023-01-05-Astaroth-Guildma-infection-traffic.pcap"
  "type" : "SELKS"
  "pkt_src" : "wire/pcap"
  "src_ip" : "172.67.197.161"
  "fileinfo" : {
    "type" : "ASCII text"
    "magic" : "ASCII text, with very long lines, with no line terminators"
    "sha256" : "c50a6c6a5d927be941369d90bb82bdcc1d200b1453cdc8b247c2912b65d5d6e0"
    "state" : "CLOSED"
    "gaps" : false
    "md5" : "54c33bdbd6f8cb7f0575a827c9594c11"
    "sha1" : "fad1a2340a54d8f8176e84573a60eb84cefbb956"
    "stored" : false
    "filename" : "/Q13hCFaXNQ64X56/lzXQFOhWzChrNh642S5/93886/Imprimir_DACTES"
    "size" : 776
```

Scirius - Showcasing Flow ID
https://github.com/StamusNetworks/SELKS

Suricata explained in one slide (IDS+NSM)

Suricata is far more than an IDS/IPS



Network Traffic
Cloud & On-premise

IDS Alerts

Protocol Transactions

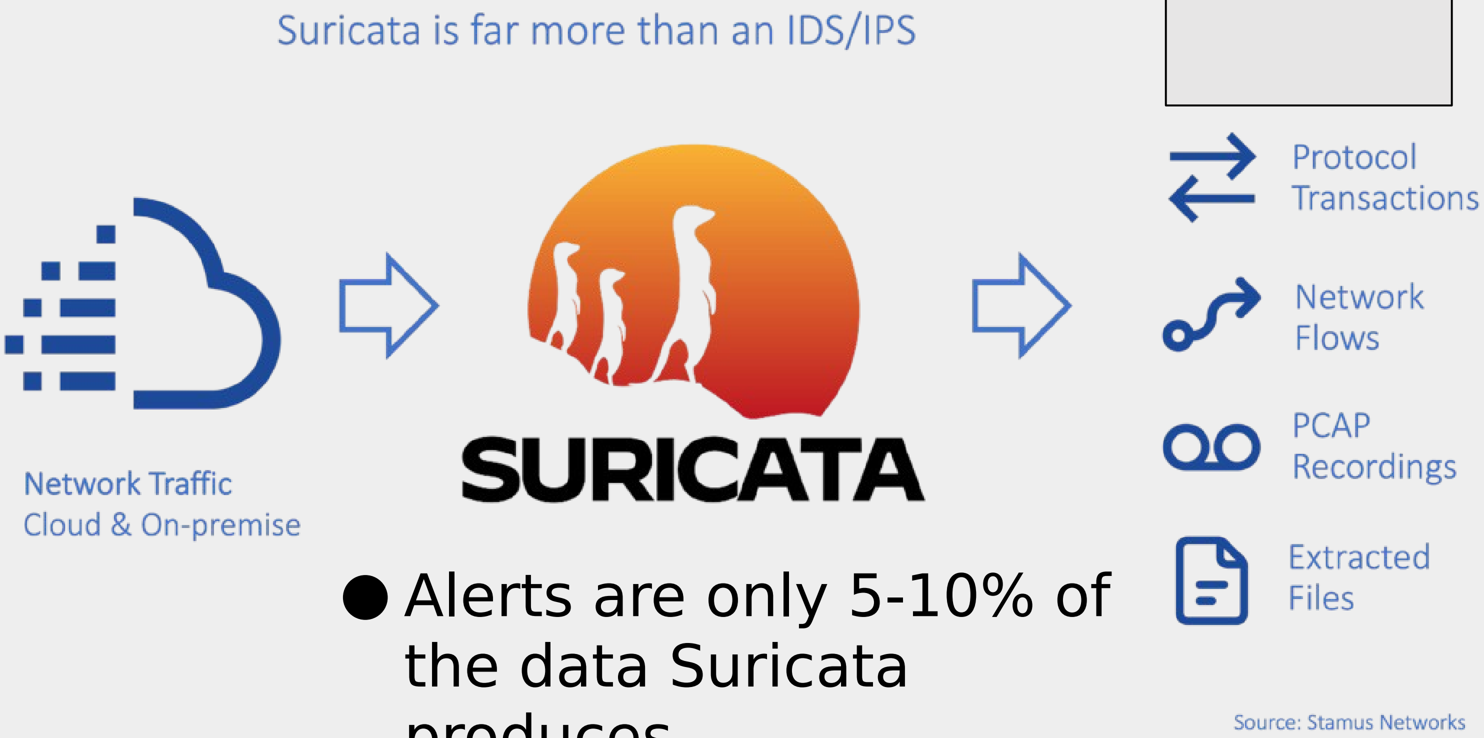
Network Flows

PCAP Recordings

Extracted Files

Source: Stamus Networks

Suricata hunting - lights/rules off (NSM)



Source: Stamus Networks

- Alerts are only 5-10% of the data Suricata produces
- Suricata works without rules too

Challenges

Adapt

Signatures evolution

- From CVE detection
  - Binary payload matching
    - Buffer overflow
    - Content triggering exploit
  - Closely bound to IPS
    - Block the payload & Protect the asset
- To …….

Signatures evolution

- To attacker behavioral analysis and infrastructure detection
  - Communication protocol characteristics (C2)
    - Type of requests (url, domain)
    - Client characteristics (used proto header, implementation)
  - Administrators behavior and process
    - TLS pattern in certificates, ...
- And notable events generation
  - Potentially interesting events: system update
  - Forensic usage

More protocol implementation

- Want to match on multiple protocols
    - Not a network grep anymore
- Want to log transaction on protocol
- Need complete support for more protocols
    - Application layer identification
        - Independently of the port
    - Application parsing
    - Application logging
    - Keyword to detect of the application player fields

Secure protocol implementation

- All protocols parser can suffer vulnerability
  - They parse the mud of internet
  - Protocols are complex
  - C language is not safe
    - Manual memory handling
- Big history of vulnerabilities on protocol parsers
  - Wireshark has a lot
  - Suricata has some too

Faster and safer implementation

- Use a combination
  - Rust: https://www.rust-lang.org/
  - Nom: https://docs.rs/nom/latest/nom/
- Rust has rich type system and ownership mode
  - Memory safety
  - Thread safety
- Nom is parser combinator library with a focus
  - on safe parsing
  - streaming patterns
  - and as much as possible zero copy.

Rust / Nom parser example

```
// PORT 192,168,0,13,234,10
named!(pub ftp_active_port<u16>,
    do_parse!(
        tag!("PORT") >>
        delimited!(multispace0, digit1, multispace0) >> tag!(",") >> digit1
>> tag!(",") >>
        digit1 >> tag!(",") >> digit1 >> tag!(",") >>
        part1: verify!(parse_u16, |&v| v <= std::u8::MAX as u16) >>
        tag!(",") >>
        part2: verify!(parse_u16, |&v| v <= std::u8::MAX as u16) >>
        (
          part1 * 256 + part2
        )
    )
);
```

Outside evolution

- Increasing network speed
  - 40G was unthinkable
  - 100G and more is the high end now
  - More traffic means more data
- Encryption
  - Less visibility
  - No more content
  - But a lot of metadata

The Challenges

- Duplicated mirror traffic
- One side async traffic
- Cloud , on prem , Virtual infrastructure
- Needs to inspect traffic regardless of RFC specs
- Encryption
- Offloading
- Monitor this ISPs 200+Gbps link
- 2 billion logs a day+ (depending on volume/size traffic)
- OS - 64 bit/32bit/arm/Linux/Windows/BSD

The Challenges

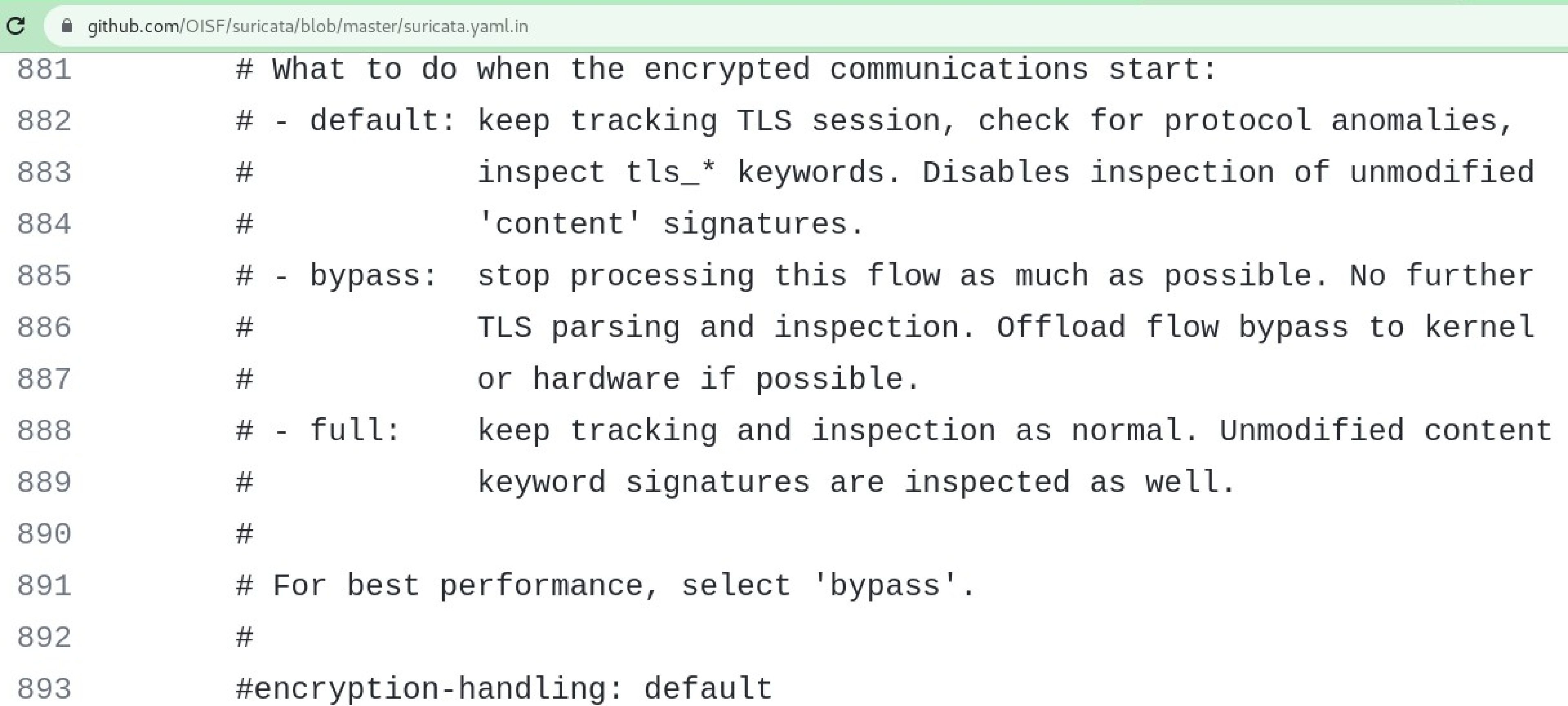
- Duplicated mirror traffic
- One side async traffic
- Cloud , on prem , Virtual infrastructure
- Needs to inspect traffic regardless of RFC specs
- Encryption
- Offloading
- Monitor this ISPs 200+Gbps link
- 2 billion logs a day+ (depending on volume/size traffic)
- OS - 64 bit/32bit/arm/Linux/Windows/BSD
- **QA anyone** ?

Encryption

All metadata is extracted during the clear text handshake:

- TLS SNI
- TLS Subject
- TLS Fingerprint
- TLS Issuer
- Certificate before/after dates
- JA3/JA3S
- TLS version

Encryption

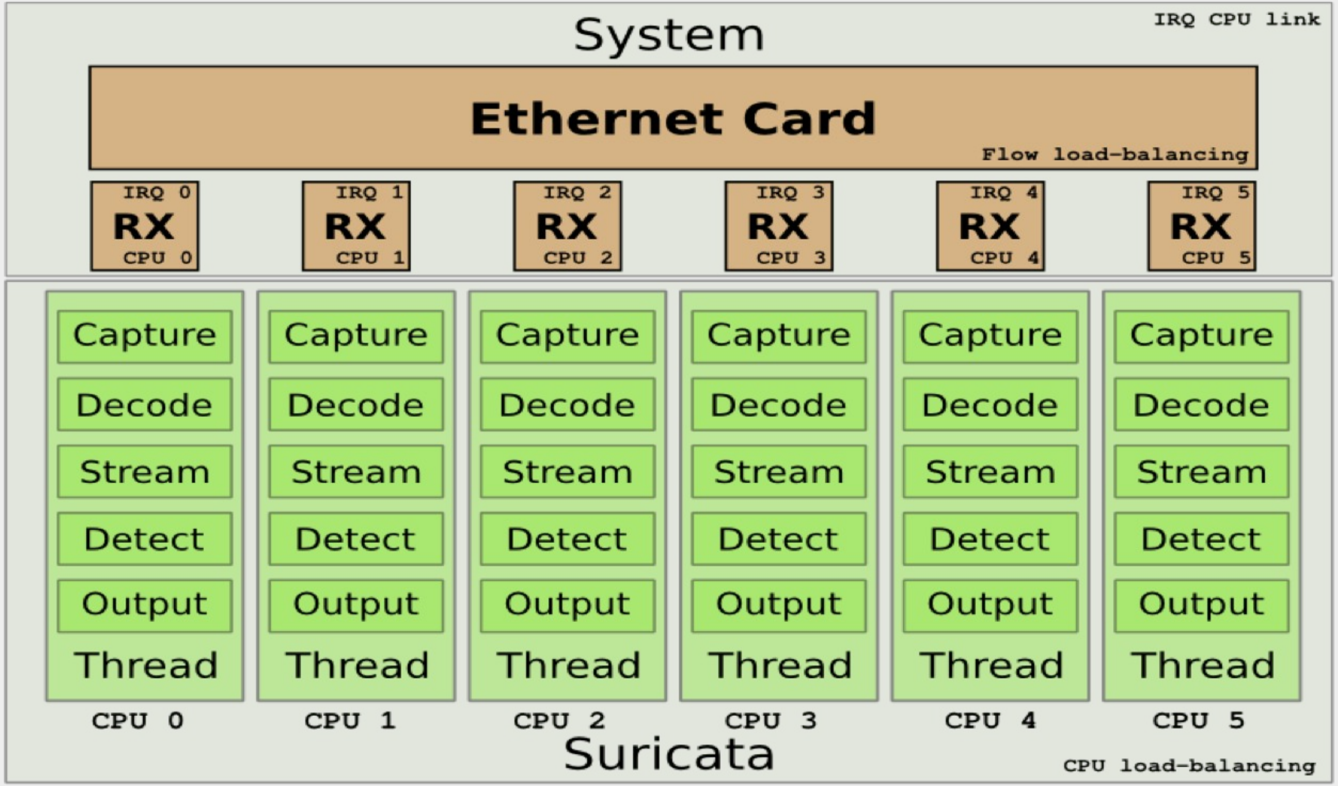


```
      # What to do when the encrypted communications start:
      # - default: keep tracking TLS session, check for protocol anomalies,
      #            inspect tls_* keywords. Disables inspection of unmodified
      #            'content' signatures.
      # - bypass:  stop processing this flow as much as possible. No further
      #            TLS parsing and inspection. Offload flow bypass to kernel
      #            or hardware if possible.
      # - full:    keep tracking and inspection as normal. Unmodified content
      #            keyword signatures are inspected as well.
      #
      # For best performance, select 'bypass'.
      #
      #encryption-handling: default
```

High performance challenges

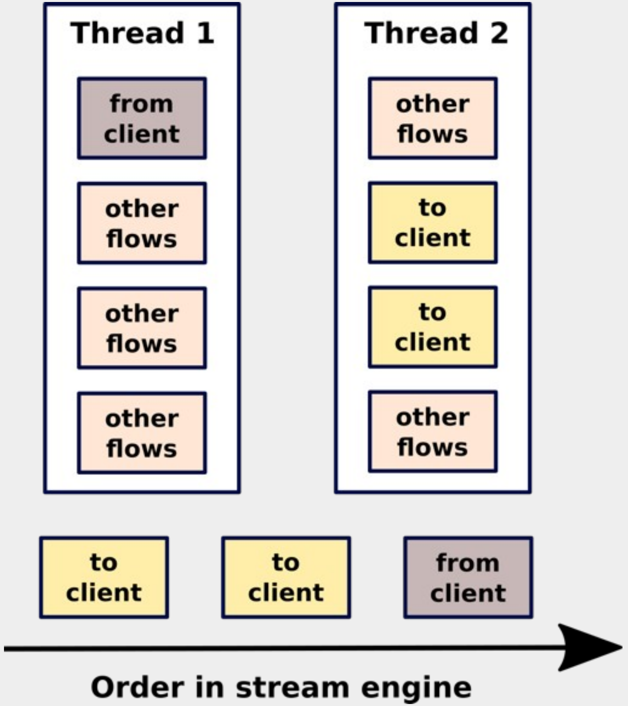
- Major perf impact factors for Suricata
  - Rules
  - Suricata version used
  - HW/OS
  - Type of traffic

Suricata - Workers mode



The RSS asymmetric hash problem

- Commodity NICs
  - Made for web/file servers to scale
  - Not build with the purpose of IDS/IPS
- IDS/IPS –needs to get both sides of a flow in the same thread, in the correct order



High performance challenges

Capture modes supported

- Netmap
- PF_RING
- AF_Packet
- AF_XDP (Suricata 7+)
- DPDK (Suricata 7+)

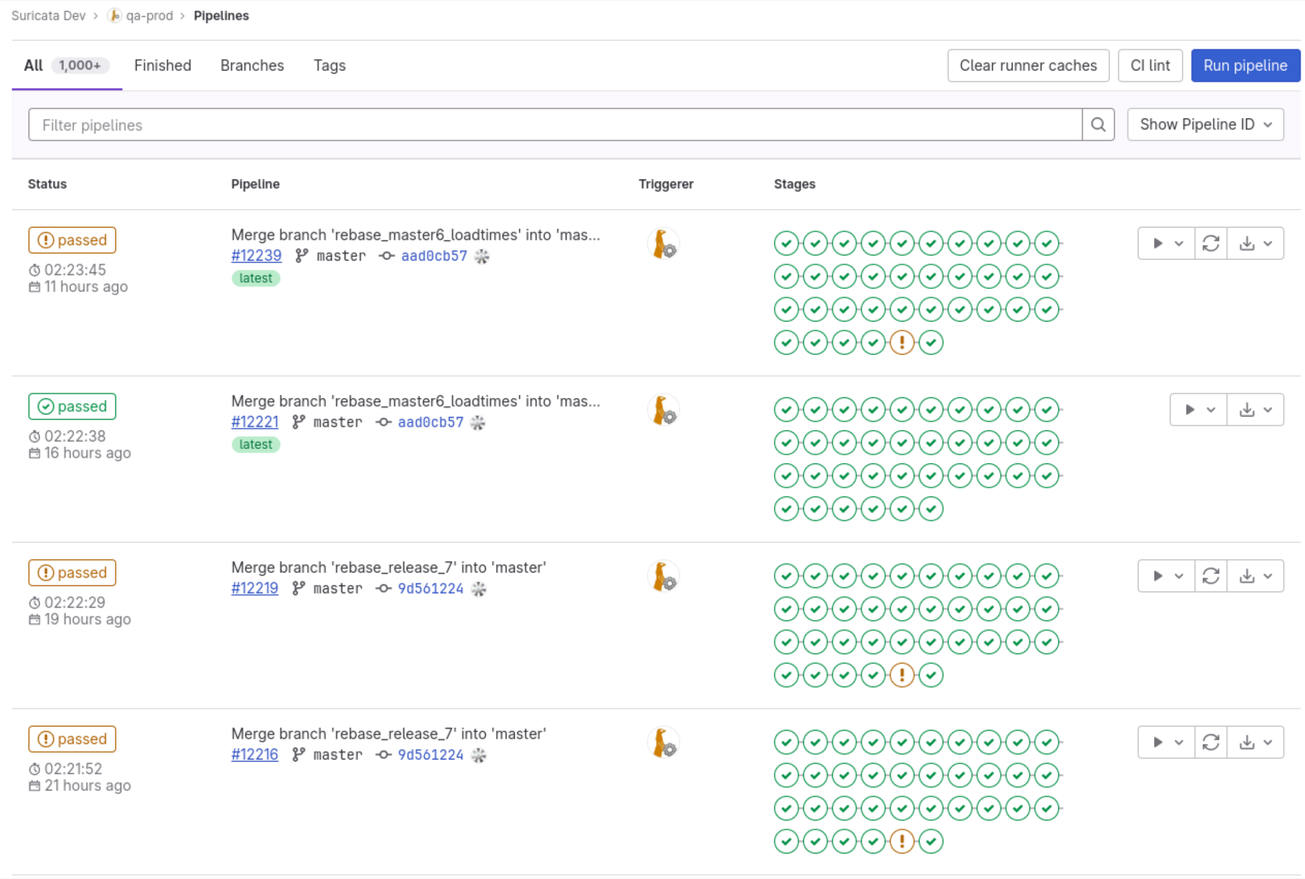
QAing Suricata

Many workflows and jobs

- Github
- Gitlab
- PPA Launchpad
- Suricata Verify
- Unit Tests
- Private runs

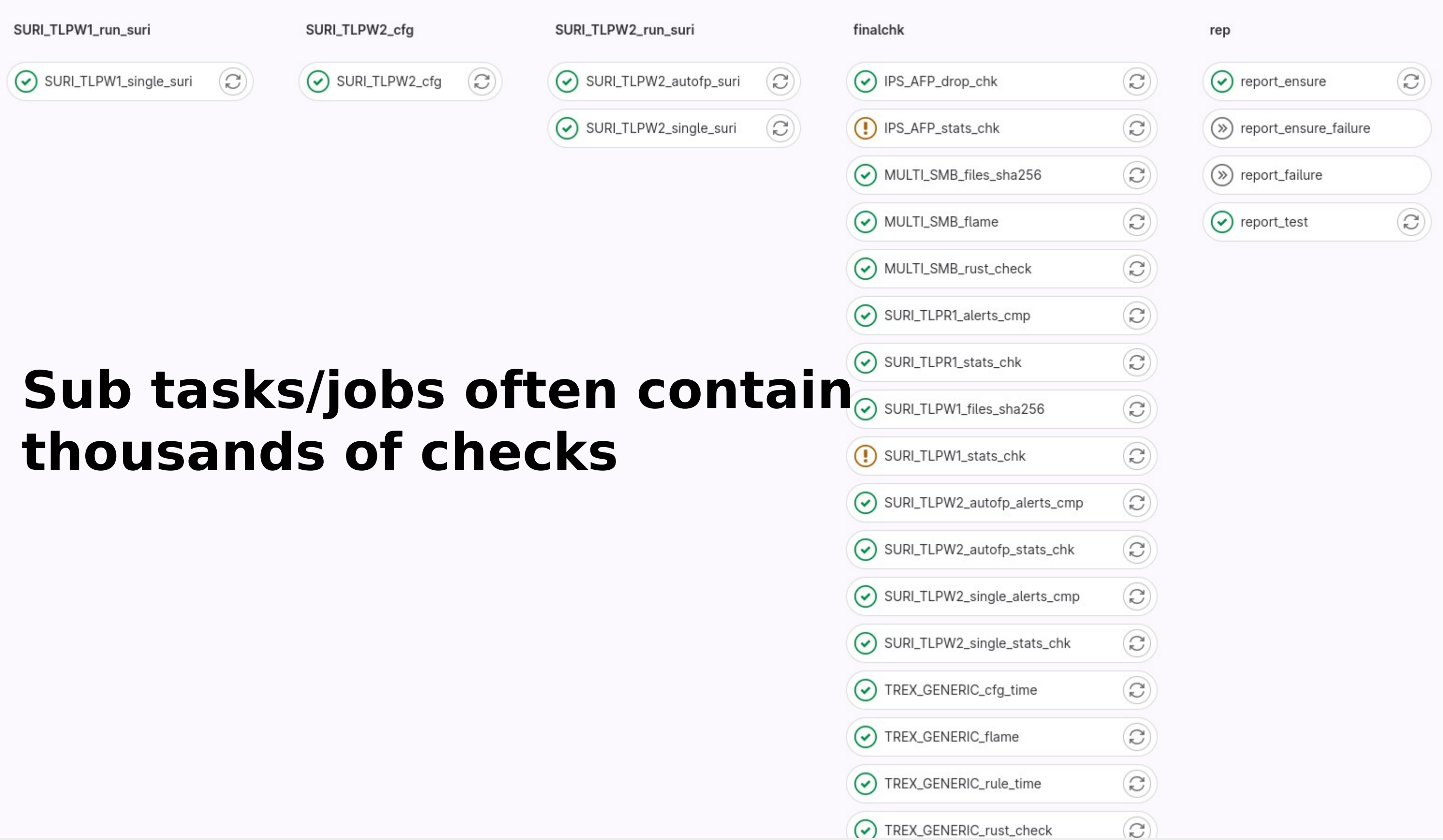
...

QAing Suricata



Status	Pipeline	Triggerer	Stages
! passed 02:23:45 11 hours ago	Merge branch 'rebase_master6_loadtimes' into 'mas... #12239 master aad0cb57 latest		✓✓✓✓✓✓✓✓✓✓ ✓✓✓✓✓✓✓✓✓✓ ✓✓✓✓✓✓✓✓✓✓ ✓✓✓✓!✓
✓ passed 02:22:38 16 hours ago	Merge branch 'rebase_master6_loadtimes' into 'mas... #12221 master aad0cb57 latest		✓✓✓✓✓✓✓✓✓✓ ✓✓✓✓✓✓✓✓✓✓ ✓✓✓✓✓✓✓✓✓✓ ✓✓✓✓✓✓
! passed 02:22:29 19 hours ago	Merge branch 'rebase_release_7' into 'master' #12219 master 9d561224		✓✓✓✓✓✓✓✓✓✓ ✓✓✓✓✓✓✓✓✓✓ ✓✓✓✓✓✓✓✓✓✓ ✓✓✓✓!✓
! passed 02:21:52 21 hours ago	Merge branch 'rebase_release_7' into 'master' #12216 master 9d561224		✓✓✓✓✓✓✓✓✓✓ ✓✓✓✓✓✓✓✓✓✓ ✓✓✓✓✓✓✓✓✓✓ ✓✓✓✓!✓

QAing Suricata



SURI_TLPW1_run_suri

- ✓ SURI_TLPW1_single_suri

SURI_TLPW2_cfg

- ✓ SURI_TLPW2_cfg

SURI_TLPW2_run_suri

- ✓ SURI_TLPW2_autofp_suri
- ✓ SURI_TLPW2_single_suri

finalchk

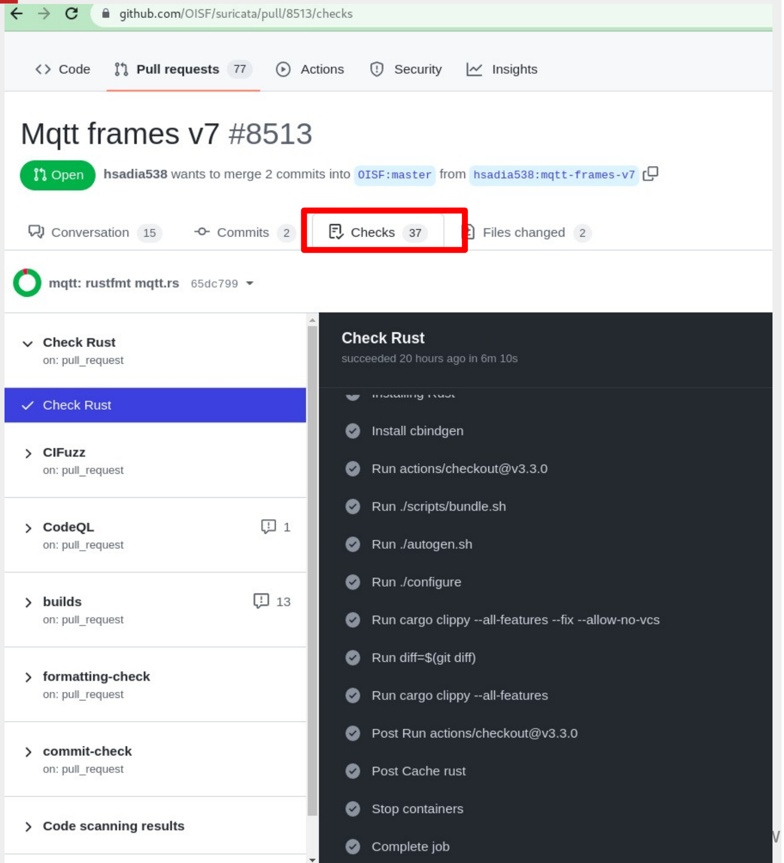
- ✓ IPS_AFP_drop_chk
- ! IPS_AFP_stats_chk
- ✓ MULTI_SMB_files_sha256
- ✓ MULTI_SMB_flame
- ✓ MULTI_SMB_rust_check
- ✓ SURI_TLPR1_alerts_cmp
- ✓ SURI_TLPR1_stats_chk
- ✓ SURI_TLPW1_files_sha256
- ! SURI_TLPW1_stats_chk
- ✓ SURI_TLPW2_autofp_alerts_cmp
- ✓ SURI_TLPW2_autofp_stats_chk
- ✓ SURI_TLPW2_single_alerts_cmp
- ✓ SURI_TLPW2_single_stats_chk
- ✓ TREX_GENERIC_cfg_time
- ✓ TREX_GENERIC_flame
- ✓ TREX_GENERIC_rule_time
- ✓ TREX_GENERIC_rust_check

rep

- ✓ report_ensure
- » report_ensure_failure
- » report_failure
- ✓ report_test

Sub tasks/jobs often contain thousands of checks

QAing Suricata



Mqtt frames v7 #8513

Open hsadia538 wants to merge 2 commits into `OISF:master` from `hsadia538:mqtt-frames-v7`

mqtt: rustfmt mqtt.rs 65dc799

Check Rust
on: pull_request

✓ Check Rust

CIFuzz
on: pull_request

CodeQL 1
on: pull_request

builds 13
on: pull_request

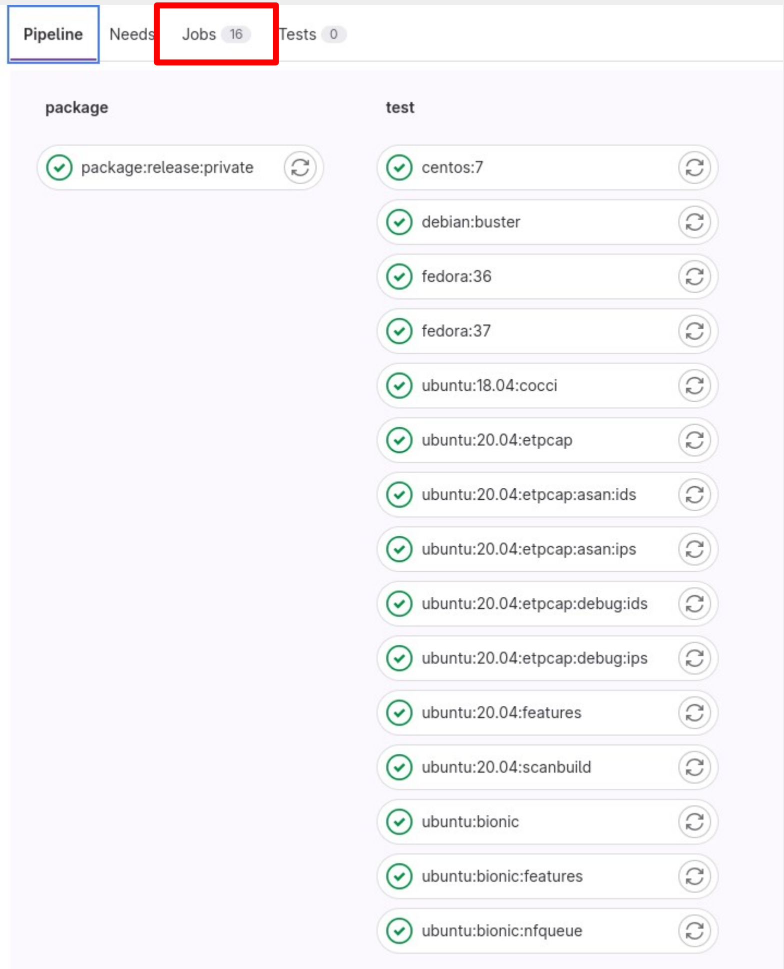
formatting-check
on: pull_request

commit-check
on: pull_request

Code scanning results

Check Rust
succeeded 20 hours ago in 6m 10s

```
Installing Rust
Install cbindgen
Run actions/checkout@v3.3.0
Run ./scripts/bundle.sh
Run ./autogen.sh
Run ./configure
Run cargo clippy --all-features --fix --allow-no-vcs
Run diff=$(git diff)
Run cargo clippy --all-features
Post Run actions/checkout@v3.3.0
Post Cache rust
Stop containers
Complete job
```



package

- package:release:private

test

- centos:7
- debian:buster
- fedora:36
- fedora:37
- ubuntu:18.04:cocci
- ubuntu:20.04:etpcap
- ubuntu:20.04:etpcap:asan:ids
- ubuntu:20.04:etpcap:asan:ips
- ubuntu:20.04:etpcap:debug:ids
- ubuntu:20.04:etpcap:debug:ips
- ubuntu:20.04:features
- ubuntu:20.04:scanbuild
- ubuntu:bionic
- ubuntu:bionic:features
- ubuntu:bionic:nfqueue

QAing Suricata

The final QA runs takes a few hours minimally, and generally runs overnight. It currently runs:

- extensive build tests on different OS', compilers, optimization levels, configure features
- static code analysis using cppcheck, scan-build
- runtime code analysis using valgrind, AddressSanitizer, LeakSanitizer
- ...

QAing Suricata

- ...
- regression tests for past bugs
- output validation of logging
- unix socket testing
- pcap based fuzz testing using ASAN and LSAN
- traffic replay based IDS and IPS tests

Contributing

Any feature or bug report can be publicly viewed and/or posted:

https://redmine.openinfosecfoundation.org/projects/suricata

How to contribute code:

https://suricata.io/2021/09/10/getting-started-contributing-to-suricata/

Current code PRs / reviews:

https://github.com/OISF/suricata/pulls

Conclusion

“It Has To Work.”

Global community effort

Needs to be open - roadmap, community discussions and input

More Resources

- Read the Docs: https://readthedocs.org/projects/suricata/
- More Suricata trainings/webinars: https://suricata.io/learn/
- Youtube: https://www.youtube.com/@OISFSuricata/videos
- Forums: https://forum.suricata.io/
- **Awesome Suricata** links: https://github.com/satta/awesome-suricata
- **Discord chat**: https://discord.com/invite/t3rV2x7MrG