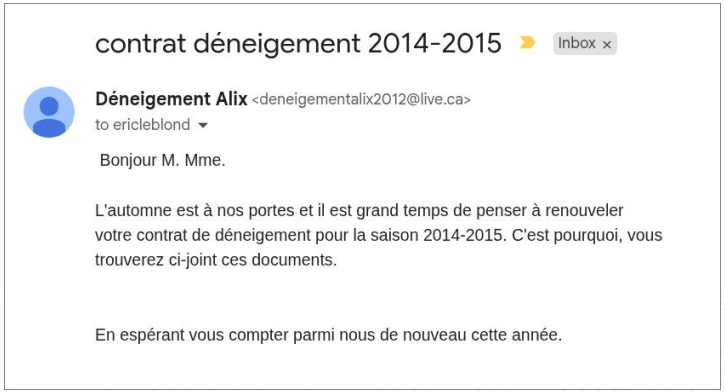
Suricata and IOCs, latest news on a love story

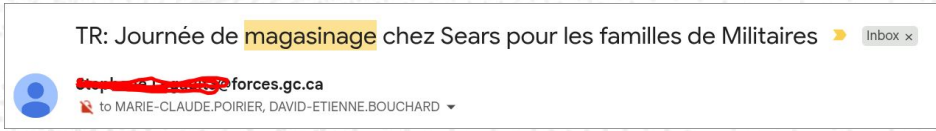
Building modern network detection with Suricata

Eric Leblond

- Suricata contributor since 2010
- OISF Board of Directors
- Co founder & CTO at Stamus Networks
- Emeritus member of Netfilter coreteam
- Suricata Language Server author
- @regit@infosec.exchange

French and first Eric Leblond on Gmail





STAMUS NETWORKS

Founded: 2014 by Éric Leblond and Peter Manev

Headquarters: Indianapolis, USA and Paris, France

Website: www.Stamus-Networks.com

Open-Source Solutions:

- *Clear NDR Community* – open source Suricata turnkey distro
- *Stamus Networks App for Splunk* – integration of SSP and Suricata into Splunk
- *Suricata Language Server* – open source syntax checking and performance guidance for Suricata rules

Commercial Solutions:

- *Clear NDR Enterprise* – commercial network threat detection, hunting and response solution

The Security Analyst's Guide To Suricata

- Dedicated to Suricata data usage
  - Hunting with Application layers logs
  - Writing signatures
  - Not a replacement of manual
- Open Source: https://github.com/StamusNetworks/suricata-4-analysts

Introduction to Suricata

The Dutch and European Army Knife for network analysis

STAMUS NETWORKS

Network Traffic
Cloud & On-premise

IDS Alerts

Protocol
Transactions

Network
Flows

PCAP
Recordings

Extracted
Files

Source: Stamus Networks

Victor Julien & Peter Manev (10 years ago)

Suricata

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

LICENSING MODEL:

General Public License v2
Or
Proprietary license

Suricata is licensed under **GPL v2**

Ensures the software remains **free, open, and community-driven**

Guarantees freedom to run, study, modify, and share improvements

👉 FAQ on GPL: suricata.io/gpl-faqs

Non technical Suricata challenges

- A huge variety of users
  - **Mega corporations**: Google, Facebook, AWS, GM, …
  - **Makers**: Hobbyist, Universities, Researchers, Trainers
  - Sandboxing: AnyRun, Triage, Joe's, …
  - **Integrated appliances**: OPNsense, pfSense, …
  - Open source IDS/NDR: Security Onion, Clear NDR Community, …
  - **NDR vendors**: Stamus Networks, Corelight, Vectra, Cylera, Neox Networks, …
  - Other vendors: Juniper, Proofpoint, Verizon
- Keep **community** happy and engaged
- Keep **consortium members** happy and engaged
  - Not all consortium members require a commercial license

Technical Suricata challenges

- High speed:
  - **100Gbps** starts to be common
  - **400Gbps** on a single server
  - But also need to run on your **raspberry**
- Work properly in real life environment
  - Input
    - Traffic capture
    - Respecting **RFC is for the weak**
    - Threat intel ingestion
  - Output
    - Export data to data lake
    - Get maximum context to the users

Release Cycle & Security Advisory

- Release Cycle
  - One major version every 2 years
  - Version being supported for 3 years
  - On average about one patch release every 2 months
- EOL Policy: https://suricata.io/our-story/eol-policy/
- Security Policy: https://github.com/OISF/suricata/security

How we got to Suricata 8.0

2+ years of development

- Total: **2090** files changed, **160167** insertions(+), **112455** deletions(-)
- **Rust**: 470 files changed, 62110 insertions(+), 9244 deletions(-)
- C: 1294 files changed, 70313 insertions(+), 98235 deletions(-)
- **Docs**: 158 files changed, 14237 insertions(+), 2862 deletions(-)

👉 Check Out: Suricata 8.0 Release Announcement

Suricata & IOCs: first contacts

History of Indicators of Compromises in Suricata land

STAMUS NETWORKS

Short history of Suricata IoCs handling

- IP only rules
  - Performant way to match on IP addresses
- IPrep
  - First file based approach
  - Signature use iprep keyword
    - Points to definition
    - IPrep definition done in suricata.yaml
- Dataset
  - List matching for sticky buffer keywords

Dataset concept

- List matching on sticky buffer keywords
  - Match on a list on extracted metadata
  - High speed
  - Multiple dimensions
- Examples:
  - alert http … (http.user_agent; dataset:isnotset,official_agents;)
  - alert smb … (file.name; content:".exe"; endswith; dataset:isnotset,good_exe;)

Problematic

- Context does matter
  - Signature match different values
  - Are all the values for same
    - Severity
    - Threat actor
- Let’s take an example:

alert http any any => any any (msg:”interesting-content”;

http.body; content: “tabernacle”;

http.host; dataset:isset,fq, load france-quebec-website.lst, type string;

…)

Tabernacle in French French

- Tabernacle is the eucharistic cupboard
- Never used in daily conversation
- If signature match on French website:
  - Article on religion
  - French christians talking about the mass

Tabernacle in Canadian French

- Tabernacle (pronounce Tabernak) is used as a **bad** swearing word.
- Quebec French profanity - Wikipedia https://share.google/sYESMkQHVJYoE1X8k
- If signature match on Quebec website:
  - Possible angry Quebec person

Resulting alert

- Contains alert meta information
  - Nothing about the website nationality
- Contains HTTP information
  - URI
  - Host
  - User-Agent
  - But nothing about website nationality
- If only we knew if website is from Quebec or from France

Only way to get context

- One signature per domain in set:
  - Message contains the classification
  - alert.metadata.site_origin contains it too

alert http any any **=>** any any (msg:”interesting-content (**Quebec**)”;

http.body; content: “tabernacle”;

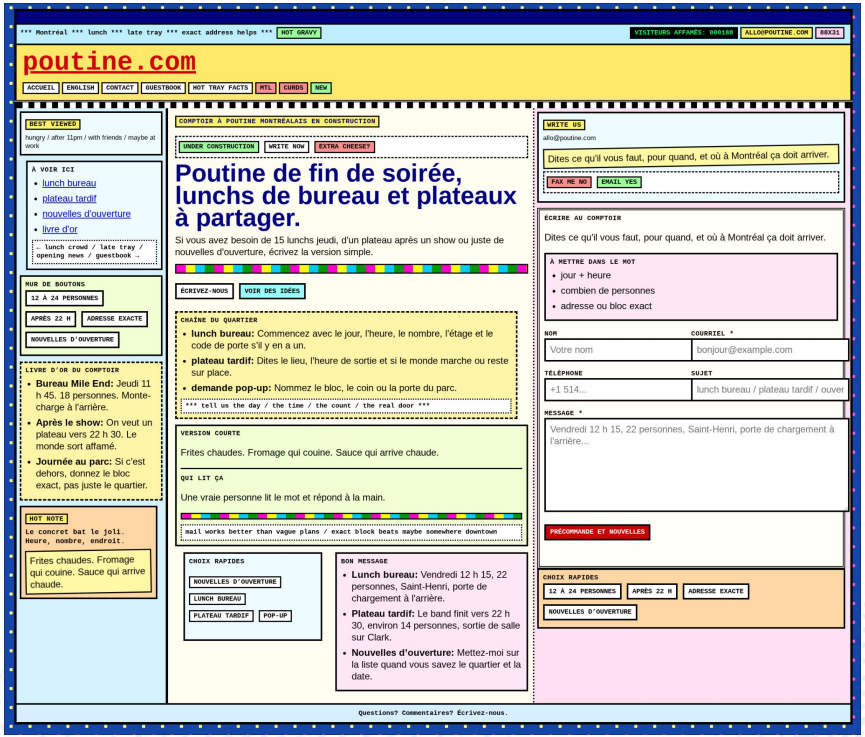
http.host; domain; content: “poutine.com”;

metadata: site_origin **Quebec**, author Stamus;

…)

- Scale up to a few 100000 domains

Result in the alert




```
"alert": {
  "action": "allowed",
  "gid": 1,
  "signature_id": 1,
  "rev": 0,
  "signature": "website (Quebec)",
  "category": "",
  "severity": 3,
  "metadata": {
    "author": [
      "Stamus"
    ],
    "site_origin": [
      "Quebec"
    ]
  }
},
"ts_progress": "client_hello_done",
"tc_progress": "server_handshake_done",
"tls": {
  "sni": "poutine.com",
  "version": "TLS 1.3",
```

Unsatisfactory solution

- Not efficient in term of
  - Performance: signature cost in memory
  - Usage: can not be easily updated
  - Import: CTI needs to know Suricata
- Not flexible:
  - Dataset has unix socket update
  - Dataset can be reused across signatures
- We need dataset with context


Introducing dataset with JSON
Formerly known as datajson
STAMVS
NETWORKS

Concept

- Embed IOCs context in the alert
- Load JSON information
- Output it in the alert
  - Under an arbitrary key
- Developed by Éric Leblond at Stamus Networks
- History
  - Been presented at Suricata 2024
    - As datajson with a different keyword
    - Using different format
  - Merged in Suricata 8.0
    - In a completely different form
    - With some goodies

Implementation in Suricata 8.0

- Evolution of dataset keyword
  - Options added to the keyword
  - Code base linked
- 2 entry formats for data file
  - JSON:
    - file is a complete JSON object
    - Convenient for REST API call return
  - New line Delimited JSON
    - Convenient for streamed approach

Demonstration

Relax and enjoy the ride

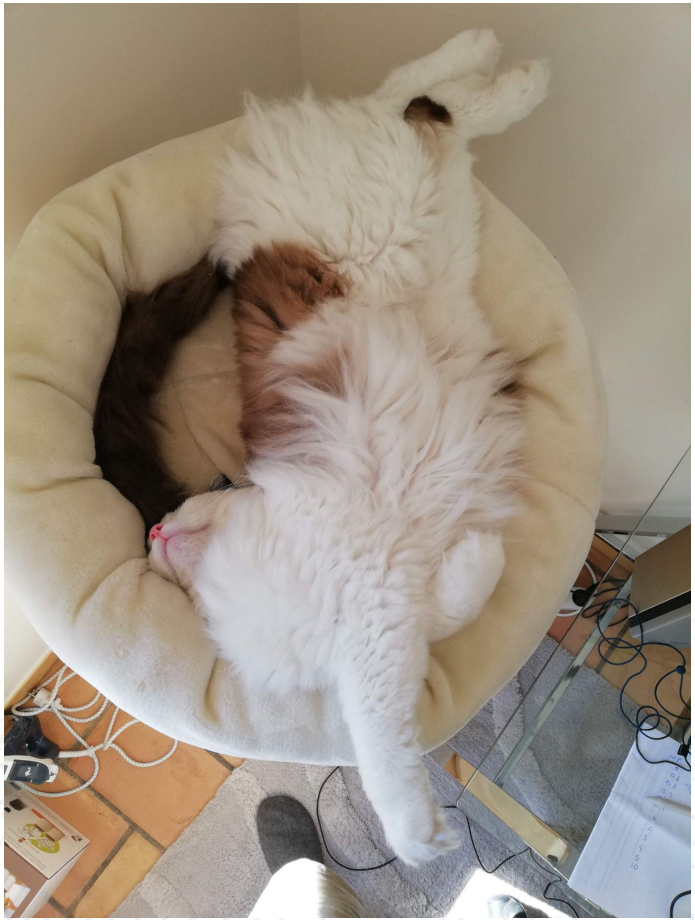
- Pcap from Malware Traffic Analysis
- Fake IOCs in NDJson format
- Signatures for multiple protocols

Unix socket interaction

- Suricata unix socket
  - Interact with Suricata live
  - Via a JSON based protocol over a unix socket
- Dataset has some dedicated commands
  - Add an element
  - Remove an element

Demonstration

- Multiple pcap run via unix socket
- One file name matching signature
- Build the IOC list from parsing
  - Via unix socket commands



Canadian Goodie: PCRE group output

- Thanks Jason Ish for the concept
- Output extracted group into the alert
- Method:
  - Name a capture “group alert_(*)”
  - Get the result in “alert.context.(*)”
- Example:
  - pcre:"/CN=(?P<alert_subject_cn>[a-zA-Z\.]+)/"
  - Create alert.context.subject_cn field

Demonstration

- Example:

alert tls any any -> any any (msg:"tls ou"; flow:established,to_client; tls.cert_subject; content:"CN="; pcre:"/CN=(?P<alert_subject_cn>[a-zA-Z\.]+)/"; sid:1000001; rev:1;)

- Output:

$ cat /tmp/eve.json|jq 'select(.alert)|.alert.context.subject_cn'

ecs.office.com

Integration with MISP

STAMUS NETWORKS

Work in progress

- Discussion with MISP team in progress
- With objective:
  - JSON export of IOCs
- Current status
  - Need to write an export script
  - See Suricon 2024 for example

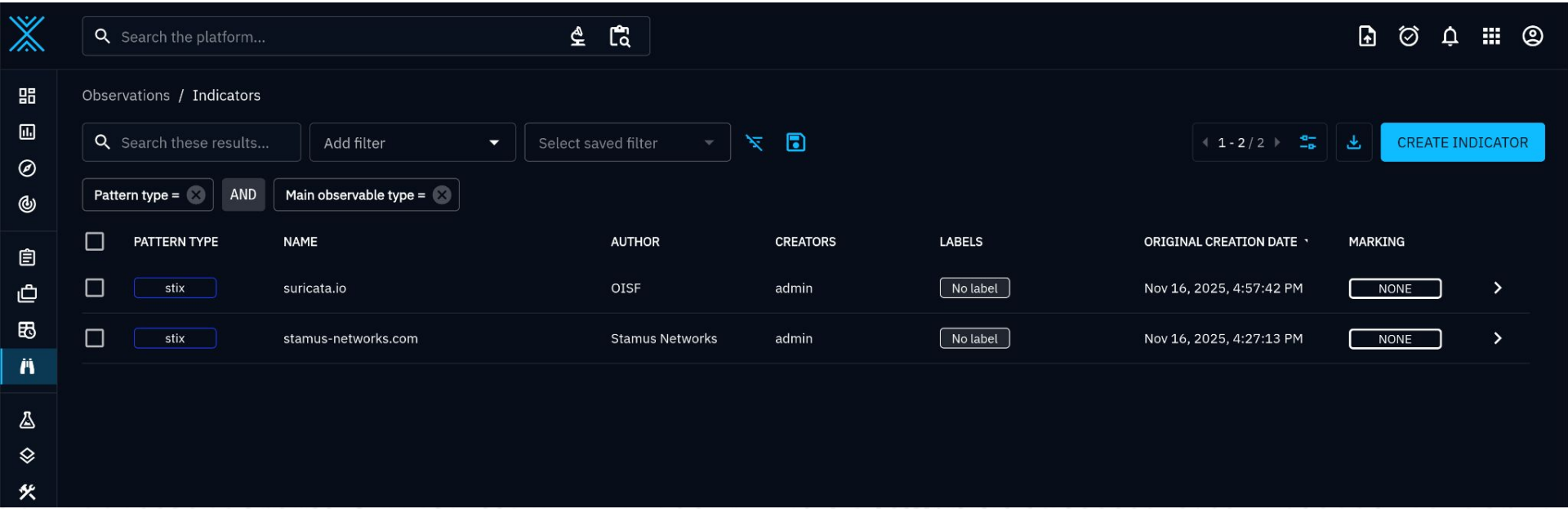
Integration with OpenCTI

STAMUS NETWORKS

Open CTI: The Threat Intelligence Platform

- **Centralized CTI Repository**: A single platform to ingest, structure, and contextualize all Cyber Threat Intelligence (CTI).
- **STIX 2.1 Standard**: Uses the STIX 2.1 data standard for interoperability and seamless exchange with other security tools.
- **Relationship Mapping**: Models complex connections between IOCs, TTPs, and Threat Actors for visual attack mapping.
- **Automation via Connectors**: Integrates with external sources (MITRE, MISP, etc.) to automatically enrich and update threat data.
- **Visual Analysis & Context:** Provides graphic tools for analysis, linking threat intelligence to the MITRE ATT&CK framework.

Export


Search the platform...
Observations / Indicators
Search these results...
Add filter
Select saved filter
1 - 2 / 2
CREATE INDICATOR
Pattern type =
AND
Main observable type =
PATTERN TYPE
NAME
AUTHOR
CREATORS
LABELS
ORIGINAL CREATION DATE
MARKING
stix
suricata.io
OISF
admin
No label
Nov 16, 2025, 4:57:42 PM
NONE
stix
stamus-networks.com
Stamus Networks
admin
No label
Nov 16, 2025, 4:27:13 PM
NONE

Export


```
  {
    "id": "identity--161d09db-3f3e-54c4-b4de-d4c85f33583a",
    "spec_version": "2.1",
    "identity_class": "organization",
    "name": "Stamus Networks",
    "created": "2025-11-16T15:26:02.491Z",
    "modified": "2025-11-16T15:26:02.491Z",
    "x_opencti_id": "ad2127a3-7e18-4cba-9b18-58fbe1d92e32",
    "x_opencti_type": "Organization",
    "type": "identity"
  },
  {
    "id": "indicator--3664f607-5b9a-582c-9583-179abd8f676f",
    "spec_version": "2.1",
    "revoked": false,
    "confidence": 100,
    "created": "2025-11-16T15:27:13.586Z",
    "modified": "2025-11-16T15:27:13.601Z",
    "pattern_type": "stix",
    "pattern": "[domain-name:value = 'stamus-networks.com']",
    "name": "stamus-networks.com",
    "description": "Stamus Networks main domain.",
    "valid_from": "2025-11-16T15:27:13.559Z",
    "valid_until": "2026-02-27T06:22:21.819Z",
    "x_opencti_score": 50,
    "x_opencti_detection": false,
    "x_opencti_main_observable_type": "Domain-Name",
    "x_opencti_observable_values": [
      {
        "type": "Domain-Name",
        "value": "stamus-networks.com",
        "hashes": null
      }
    ],
    "x_opencti_id": "c3b6963a-975a-4da6-9fdc-3be7a570fbb8",
    "x_opencti_type": "Indicator",
    "type": "indicator",
    "created_by_ref": "identity--161d09db-3f3e-54c4-b4de-d4c85f33583a"
  }
]
```

Transform

- Export contains indicators and side information
- Need to remove the non indicators:

```
cat opencti-indicators.rules|jq '.objects |= map(select(.type=="indicator"))'
```

- May be fixable
  - In OpenCTI export
  - In Suricata by adding a test to exclude entry ?

Signatures

alert **http** any any -> any any (msg:"OpenCTI Detected"; http.host; **domain**; **dataset**:isset,opencti,type string,load opencti-ioc-domain.json,format json,context_key opencti,array_key objects,value_key name; sid:1000001; rev:1;)

alert **tls** any any -> any any (msg:"OpenCTI Detected"; tls.sni; **domain**; **dataset**:isset,opencti,type string,load opencti-ioc-domain.json,format json,context_key opencti,array_key objects,value_key name; sid:1000002; rev:1;)

Event


```
"context": {
  "opencti": {
    "id": "indicator--dcf6ef40-ec92-5a4d-8eb4-2de21ba0f16f",
    "spec_version": "2.1",
    "revoked": false,
    "confidence": 100,
    "created": "2025-11-16T15:57:42.035Z",
    "modified": "2025-11-16T15:57:42.062Z",
    "pattern_type": "stix",
    "pattern": "[domain-name:value = 'suricata.io']",
    "name": "suricata.io",
    "description": "Suricata main domain",
    "valid_from": "2025-11-16T15:57:41.994Z",
    "valid_until": "2026-02-27T06:52:50.254Z",
    "x_opencti_score": 50,
    "x_opencti_detection": false,
    "x_opencti_main_observable_type": "Domain-Name",
    "x_opencti_observable_values": [
      {
        "type": "Domain-Name",
        "value": "suricata.io",
        "hashes": null
      }
    ],
    "x_opencti_id": "deb5d4ba-e199-40c0-b030-d13cbada1131",
    "x_opencti_type": "Indicator",
    "type": "indicator",
    "created_by_ref": "identity--d2a2b6ca-76cd-5c19-8b5d-fa5917824943"
  }
}
```

Conclusion

Time to pack up

Conclusion

IoCs handling in Suricata

- High performance
  - On almost all metadata
- Full of context
  - With dataset with JSON
- Dynamic
  - Unix socket can be used for edition of the list

Dataset with JSON

- Evolution of dataset
- Bring all context in alert

Questions ?

Suricata: https://suricata.io

Stamus: https://www.stamus-networks.com

Want more: Suricata Threat Detection workshop this afternoon

Bon appétit!