Distributing Security Content to Detect Threats Across Past, Present, and Future

SuriCon 2021, Boston

Sascha Steinbiß[1] **Matthias Vallentin**[2]

[1]DCSO Deutsche Cyber-Sicherheitsorganisation GmbH, Berlin, Germany
[2]Tenzir GmbH, Hamburg, Germany

About the speakers

TLP:WHITE

Sascha

- Senior Software Engineer
- 5y @DCSO, German MSS provider
- Former genome wrangler
- Suricata and Debian contributor

Matthias

- Founder & CEO at Tenzir
- PhD @ UC Berkeley (with Zeek team)
- High-performance network monitoring
- SOC infrastructure and threat detection

Definitions

TLP:WHITE

- Definitions in the context of the talk
  - Security Content (SC): Rules, IoCs, scripts/code (“Detection as Code”)
  - In this talk: Suricata rules and tactical indicators
- How can we instrumentalize rules and TI for detection?
  - Rules: used and shared directly
  - Indicators:
    - wrapped in rules
    - datasets
    - matched downstream
  - Code: Rules + Lua, (Zeek Scripts) heavy-weight for sharing

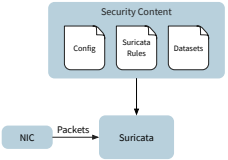
```
# Rule
alert smb $HOME_NET any -> any any \
(msg:"ET EXPLOIT Possible ETERNALBLUE \
MS17-010 Echo Response";
flow:from_server, established; \
content:"|00 00 00 31 ff|SMB|2b ... 07 c0|"; \
depth:16; fast_pattern; [...]
sid:2024218; rev:2;)

# Rule with embedded IoC
alert dns any any -> any any \
(msg:"DNS Query to evil.com"; \
dns.query; content:"evil.com"; depth:8; \
fast_pattern; nocase; endswith; \
sid:1000023; rev:1;)

# Rule with dataset reference
alert dns any any -> any any \
(msg:"DNS Query to evil domain list"; \
dns.query; dataset:isset,evil-dns;
sid:1000042; rev:1;)
```

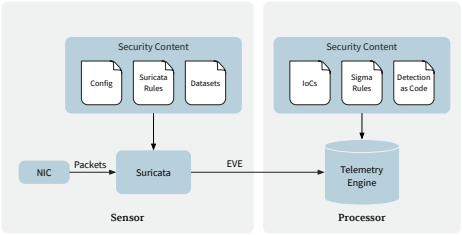
Off-the-Shelf Suricata Deployment

TLP:WHITE



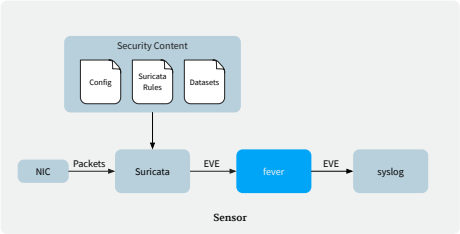
Off-the-Shelf Suricata Deployment

TLP:WHITE



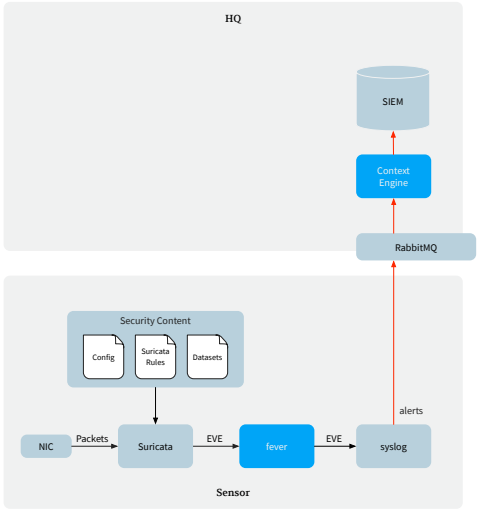
Off-the-Shelf Suricata Deployment

TLP:WHITE



Off-the-Shelf Suricata Deployment

TLP:WHITE



Security Content Management (SCM) with Suricata

TLP:WHITE

Native

- Rules
  - via pull action (e.g. via `suricata-update`)
  - Suricata reload (e.g. `suricatasc`)
- Datasets
  - distribution not handled at all by current tooling – Download via `curl`?
  - Suricata reload (e.g. `suricatasc`)?

Downstream

- Bloom filters
  - Download via `curl`
  - FEVER reload via CLI or gRPC
- Sigma rules
  - SIEM-specific (periodic) triggering on logs

Security Content Management Challenges

TLP:WHITE

1. Rule updating is the only supported method by the “default tooling”
   - No best practices on how to update datasets
   - Everything beyond rules happens downstream (not standardized)
2. Inefficient bulk security content download
   - Traffic: feasible in low-bandwidth environment?
   - Granularity: is it worth sending updates for just one new IoC?
3. Roll out takes time and causes delay
   - Inevitable lower bound: time passes until rule can fire on sensor
   - Multiple steps involved
   - Trade-off: time to detection vs. overhead of rule reload
4. Rolling out content in a multi-sensor environment (MSSP, large corp)
   - Scalable rule dissemination to a large sensor fleet
   - Sensors may need different subsets of content

Addressing SCM Challenges

TLP:WHITE

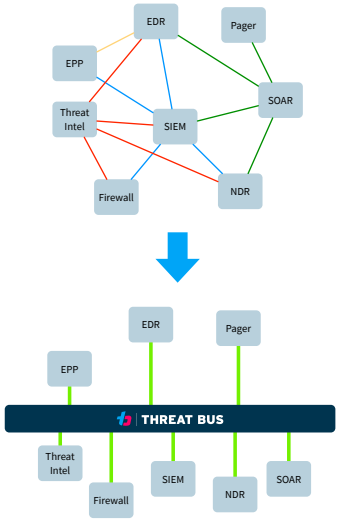
- Decouple expression of security content from application in Suricata
  - Ingress: many different formats of SC to consume (BLs, STIX feeds, etc.)
  - Egress: many ways to apply SC (Rules, datasets, etc.)
  - Have a common (standardized?) carrier data format
- Have a unified delivery path for security content
  - Consume SC from various sources (APIs, web sites, files)
  - Fine-grained updates, as opposed to simple bulk transfers
  - Real-time content publishing, as opposed to periodic polling
  - Publish/subscribe content distribution architecture

Threat Bus

TLP:WHITE

A message broker for real-time exchange of security content

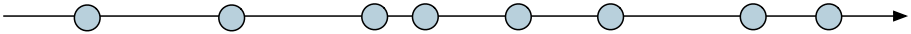
- Standardized data plane: STIX 2.1
- Plugin based
  - Backbone: underlying transport channel (AMQP, Kafka, . . .)
  - Apps: format and tool-specific connectors (Suricata, MISP, . . .)
- Communication
  - Topics vs. snapshots
- Key benefits
  - Reduce tool integration complexity
  - Vendor-neutral communication across tools
  - Quick onboarding by re-using existing messaging infrastructure



Forward Matching

TLP:WHITE

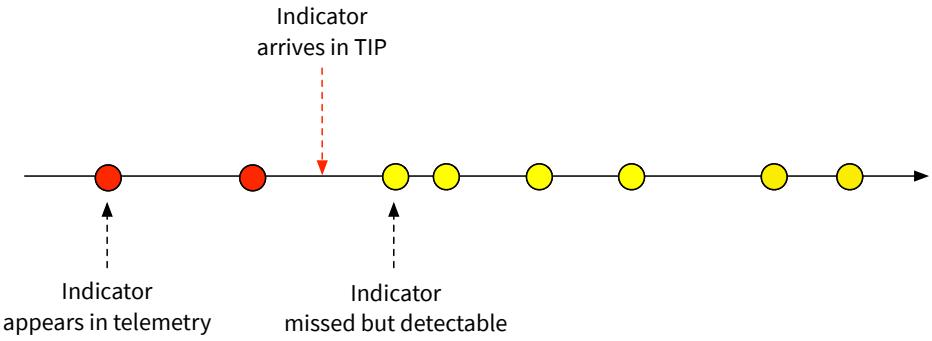
- Goal: use indicators for realtime detection as soon as they arrive
- Mechanism: apply delivered indicators in their native tool as soon as they arrive



Forward Matching

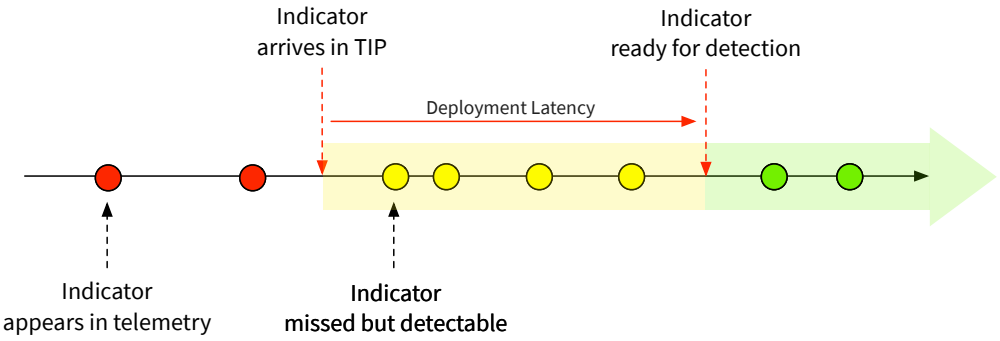
TLP:WHITE

- Goal: use indicators for realtime detection as soon as they arrive
- Mechanism: apply delivered indicators in their native tool as soon as they arrive



Forward Matching

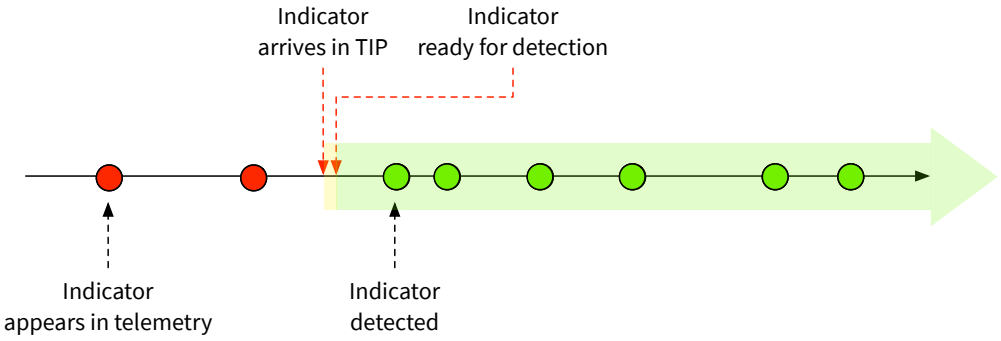
- Goal: use indicators for realtime detection as soon as they arrive
- Mechanism: apply delivered indicators in their native tool as soon as they arrive



Forward Matching

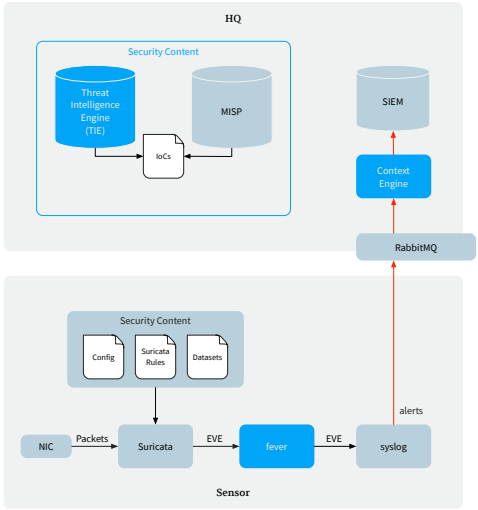
TLP:WHITE

- Goal: use indicators for realtime detection as soon as they arrive
- Mechanism: apply delivered indicators in their native tool as soon as they arrive



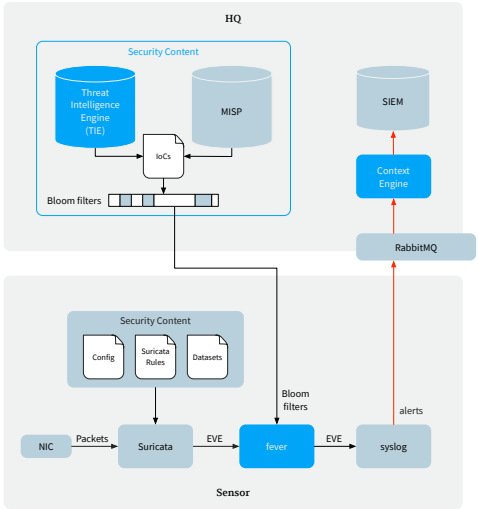
Forward Matching Architecture (I)

TLP:WHITE



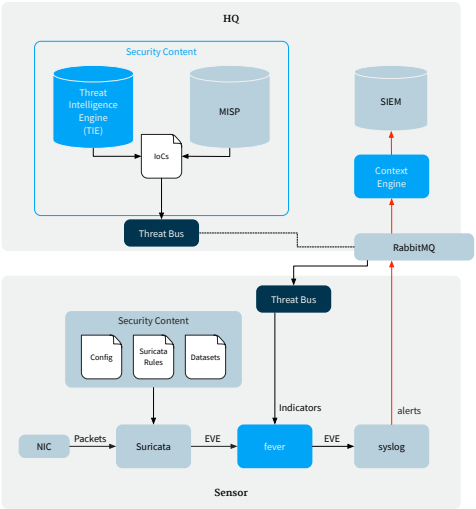
Forward Matching Architecture (II)

TLP:WHITE



Forward Matching Architecture (III)

TLP:WHITE

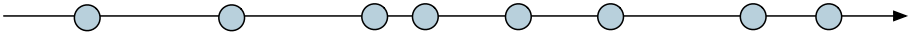


Forward Matching Demo

Problem: Indicator in the Past

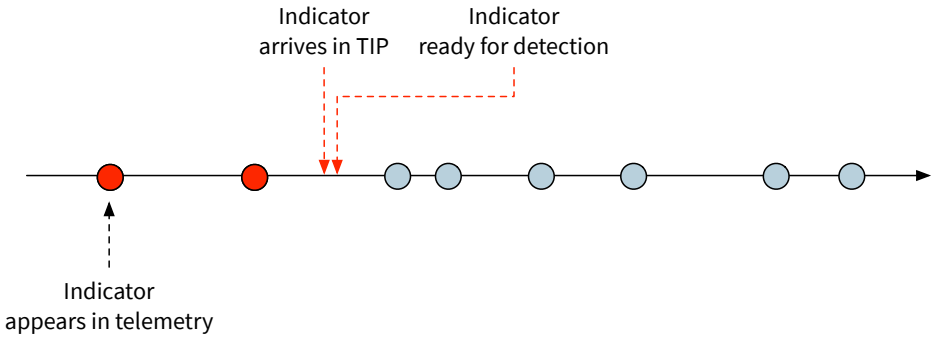
TLP:WHITE

- Goal: use indicators for realtime detection as soon as they arrive
  - Advanced attacker use initial vector only once
  - Stay long under the defenders' radar (dwell time $\approx$ 6 months)
  - Typical IR questions: how long has the attack been going on?
  - Impossible to detect with forward matching



Backward Matching

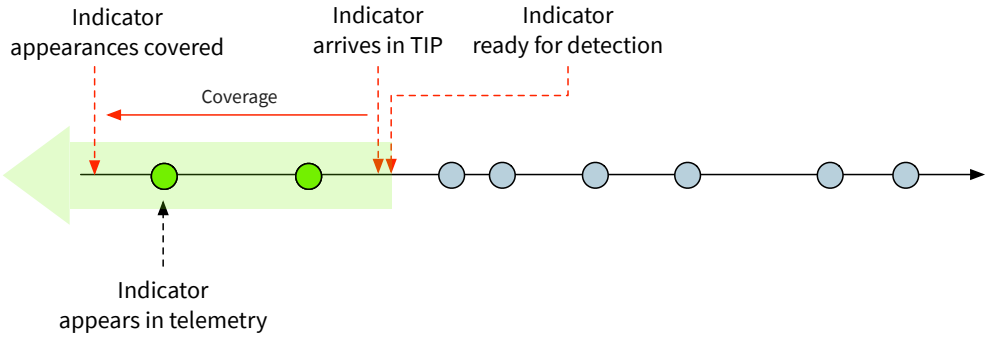
TLP:WHITE



- Solution: backward matching
- Conceptually simple: SIEM search

Backward Matching

TLP:WHITE



- Solution: backward matching
- Conceptually simple: SIEM search

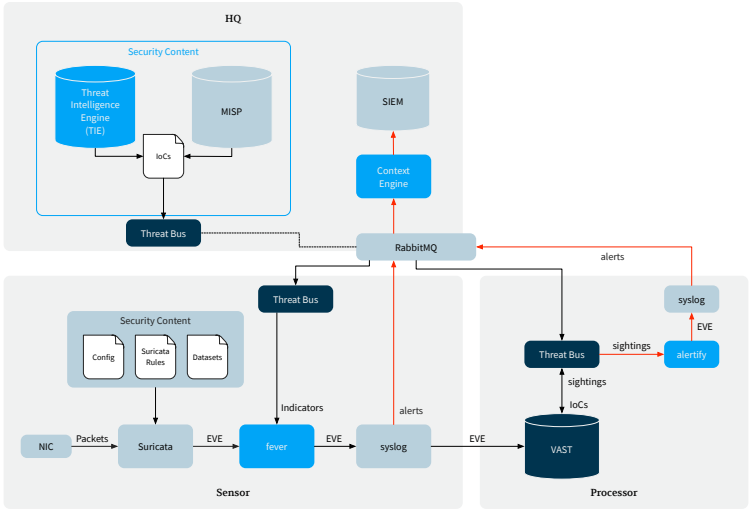
Security Content Management Challenges

TLP:WHITE

- Transparency: make live and retro alerts look identical for analysts
  - Fever: unified EVE JSON
- Automation: trigger searches automatically (across sensor fleet)
  - Threat Bus via RabbitMQ backbone
- Scalability: low-latency search && large number of queries
  - VAST (see SuriCon 2019)
- Retention: span at least attacker dwell time
  - Compaction (coming soon in VAST)

Backward Matching Architecture

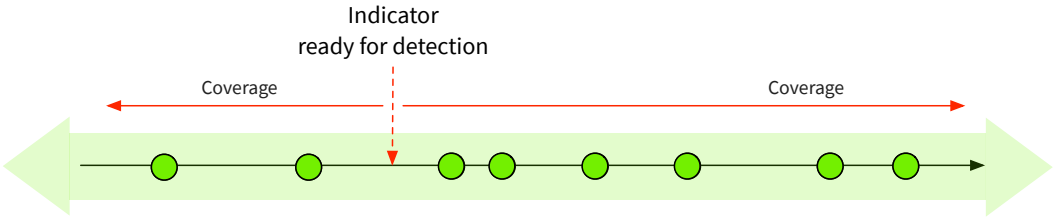
TLP:WHITE



Backward Matching Demo

Result Situation

TLP:WHITE



Summary

- Security content matching in Suricata telemetry poses various challenges
  - Distribution
  - Matcher update
  - Historical data
- Solution
  - Forward matching assisted by Threat Bus and FEVER
  - Backward matching assisted by Threat Bus and VAST
- Everything available as Free Software

Tenzir Plug

TLP:WHITE

If you don't want to re-build this yourself, we can help with:

- Build SOCs using open tools
- Operationalize threat intelligence
- Enable advanced threat detection use cases

Commercially available VAST plugins

- High-speed **IoC matching** (FEVER for all data formats)
- **Passive inventorization** using EDR and NDR telemetry
- **Compaction**: incremental aging of events for long retention/detection windows
- **NetFlow parser** for v5, v9, v10 (IPFIX)

Future Work

- Generalize Threat Bus data format:
  - STIX bundles containing SCOs
  - More structure ⇒ better response
  - Generate stateful rules
    - e.g. first lookup of 8.8.8.8 followed by HTTP GET to evil.com
- Publish Suricata Rules on topic for indicator SDO
  - Pattern-type: `suricata`
  - Topic: `stix2/indicator/suricata`
- Use Threat Bus for configuration distribution
  - e.g. dynamic update of address and port groups

```
HTTP_SERVERS: "$HOME_NET"
SMTP_SERVERS: "$HOME_NET"
SQL_SERVERS: "$HOME_NET"
DNS_SERVERS: "$HOME_NET"
TELNET_SERVERS: "$HOME_NET"
AIM_SERVERS: "$EXTERNAL_NET"
DC_SERVERS: "$HOME_NET"
DNP3_SERVER: "$HOME_NET"
DNP3_CLIENT: "$HOME_NET"
MODBUS_CLIENT: "$HOME_NET"
MODBUS_SERVER: "$HOME_NET"
ENIP_CLIENT: "$HOME_NET"
ENIP_SERVER: "$HOME_NET"
```

STIX 2.1 — SCOs vs. SDOs

TLP:WHITE

```
{
  "type": "domain-name",
  "spec_version": "2.1",
  "id": "domain-name--3c10e93f-798e-5a26-a0c1-08156efab7f5",
  "value": "evil.com"
}

{
  "type": "indicator",
  "spec_version": "2.1",
  "id": "indicator--8e2e2d2b-17d4-4cbf-938f-98ee46b3cd3f",
  "created": "2016-04-06T20:03:48.000Z",
  "modified": "2016-04-06T20:03:48.000Z",
  "indicator_types": ["malicious-activity"],
  "name": "Evil Domain",
  "pattern": "[ domain-name:value = 'evil.com' ]",
  "pattern_type": "stix",
  "valid_from": "2016-01-01T00:00:00Z"
}
```

Thanks for your attention!

Backup Slides

Threat Bus Architecture

TLP:WHITE