Community ID

Standardized flow hashing for your NSM tools

Christian Kreibich
christian@corelight.com
@ckreibich
The problem
Typical Suricata log entries (eve.json)

```
{
  "timestamp": "2003-05-05T07:51...",
  "flow_id": 23963675020689,
  "pcap_cnt": 10,
  "event_type": "alert",
  "src_ip": "203.241.248.20",
  "src_port": 3051,
  "dest_ip": "80.4.124.41",
  "dest_port": 80,
  "proto": "TCP",
  "alert": {
    "action": "allowed",
    "gid": 1,
    "signature_id": 9999999,
    "rev": 1,
    "signature": "PWNED!",
    "category": "Misc activity",
    ...
  }
}
```

```
{
  "timestamp": "2003-05-05T07:51...",
  "flow_id": 23963675020689,
  "event_type": "http",
  "src_ip": "203.241.248.20",
  "src_port": 3051,
  "dest_ip": "80.4.124.41",
  "dest_port": 80,
  "proto": "TCP",
  "tx_id": 0,
  "http": {
    "http_port": 0,
    "url": "/scripts/..%c1%9c../...",
    "http_method": "GET",
    "length": 0
  }
}
```
Typical Suricata log entries (eve.json)

```
{
    "timestamp": "2003-05-05T07:51...",
    "flow_id": 23963675020689,
    "pcap_cnt": 10,
    "event_type": "alert",
    "src_ip": "203.241.248.20",
    "src_port": 3051,
    "dest_ip": "80.4.124.41",
    "dest_port": 80,
    "proto": "TCP",
    "alert": {
        "action": "allowed",
        "gid": 1,
        "signature_id": 9999999,
        "rev": 1,
        "signature": "PWNED!",
        "category": "Misc activity",
        ...
    }
}

{
    "timestamp": "2003-05-05T07:51...",
    "flow_id": 23963675020689,
    "event_type": "http",
    "src_ip": "203.241.248.20",
    "src_port": 3051,
    "dest_ip": "80.4.124.41",
    "dest_port": 80,
    "proto": "TCP",
    "tx_id": 0,
    "http": {
        "http_port": 0,
        "url": "/scripts/..%c1%9c../...",
        "http_method": "GET",
        "length": 0
    }
}
```
Typical Suricata log entries (eve.json)

```
{
  "timestamp": "2003-05-05T07:51:...",
  "flow_id": 23963675020689,
  "pcap_cnt": 10,
  "event_type": "alert",
  "src_ip": "203.241.248.20",
  "src_port": 3051,
  "dest_ip": "80.4.124.41",
  "dest_port": 80,
  "proto": "TCP",
  "alert": {
    "action": "allowed",
    "gid": 1,
    "signature_id": 9999999,
    "rev": 1,
    "signature": "PWNED!",
    "category": "Misc activity",
    ...
  }
}
{
  "timestamp": "2003-05-05T07:51:...",
  "flow_id": 23963675020689,
  "event_type": "http",
  "src_ip": "203.241.248.20",
  "src_port": 3051,
  "dest_ip": "80.4.124.41",
  "dest_port": 80,
  "proto": "TCP",
  "tx_id": 0,
  "http": {
    "http_port": 0,
    "url": "/scripts/../%c1%9c/../...",
    "http_method": "GET",
    "length": 0
  }
}
```
Typical Zeek log entries

**conn.log**

```json
{
  "ts":1052146262.937361,
  "uid":"CVKjZo2GrV8DM0Fvo5",
  "id.orig_h":"203.241.248.20",
  "id.orig_p":3051,
  "id.resp_h":"80.4.124.41",
  "id.resp_p":80,
  "proto":"tcp",
  "service":"http",
  "duration":6.582984,
  ...
}
```

**http.log**

```json
{
  "ts":1052146263.269431,
  "uid":"CVKjZo2GrV8DM0Fvo5",
  "id.orig_h":"203.241.248.20",
  "id.orig_p":3051,
  "id.resp_h":"80.4.124.41",
  "id.resp_p":80,
  "trans_depth":1,
  "method":"GET",
  "uri": ...,
  "request_body_len": 0,
  ...
}
```
Typical Zeek log entries

conn.log
{
  "ts":1052146262.937361,
  "uid":"CVKjZo2GrV8DM0Fvo5",
  "id.orig_h":"203.241.248.20",
  "id.orig_p":3051,
  "id.resp_h":"80.4.124.41",
  "id.resp_p":80,
  "proto":"tcp",
  "service":"http",
  "duration":6.582984,
  ...
}

http.log
{
  "ts":1052146263.269431,
  "uid":"CVKjZo2GrV8DM0Fvo5",
  "id.orig_h":"203.241.248.20",
  "id.orig_p":3051,
  "id.resp_h":"80.4.124.41",
  "id.resp_p":80,
  "trans_depth":1,
  "method":"GET",
  "uri":...
  "request_body_len":0,
  ...
}
Typical Zeek log entries

**conn.log**

```
{  
  "ts":1052146262.937361,
  "uid":"CVKjZo2GrV8DM0Fvo5",
  "id.orig_h":"203.241.248.20",
  "id.orig_p":3051,
  "id.resp_h":"80.4.124.41",
  "id.resp_p":80,
  "proto":"tcp",
  "service":"http",
  "duration":6.582984,
  ...
}
```

**http.log**

```
{  
  "ts":1052146263.269431,
  "uid":"CVKjZo2GrV8DM0Fvo5",
  "id.orig_h":"203.241.248.20",
  "id.orig_p":3051,
  "id.resp_h":"80.4.124.41",
  "id.resp_p":80,
  "trans_depth":1,
  "method":"GET",
  "uri": ...,
  "request_body_len": 0,
  ...
}
```

*ALSO WOOT!*
Typical Zeek-and-Suricata log entries

```
{
    "ts": 1052146262.937361,
    "uid": "CVKjZo2GrV8DM0Fvo5",
    "id.orig_h": "203.241.248.20",
    "id.orig_p": 3051,
    "id.resp_h": "80.4.124.41",
    "id.resp_p": 80,
    "proto": "tcp",
    "service": "http",
    "duration": 6.582984,
    ...
}
```

```
{
    "timestamp": "2003-05-05T07:51...",
    "flow_id": 23963675020689,
    "pcap_cnt": 10,
    "event_type": "alert",
    "src_ip": "203.241.248.20",
    "src_port": 3051,
    "dest_ip": "80.4.124.41",
    "dest_port": 80,
    "proto": "TCP",
    "alert": {
        "action": "allowed",
        "gid": 1,
        "signature_id": 9999999,
        "rev": 1,
        "signature": "PWNED!",
        "category": "Misc activity",
        ...
    }
}
```
Typical Zeek-and-Suricata log entries

```json
{
    "ts": 1052146262.937361,
    "uid": "CVKjZo2GrV8DM0Fvo5",
    "id.orig_h": "203.241.248.20",
    "id.orig_p": 3051,
    "id.resp_h": "80.4.124.41",
    "id.resp_p": 80,
    "proto": "tcp",
    "service": "http",
    "duration": 6.582984,
    ...
}

{
    "timestamp": "2003-05-05T07:51...",
    "flow_id": 23963675020689,
    "pcap_cnt": 10,
    "event_type": "alert",
    "src_ip": "203.241.248.20",
    "src_port": 3051,
    "dest_ip": "80.4.124.41",
    "dest_port": 80,
    "proto": "TCP",
    "alert": {
        "action": "allowed",
        "gid": 1,
        "signature_id": 9999999,
        "rev": 1,
        "signature": "PWNED!",
        "category": "Misc activity",
        ...
    }
}
```
Typical Zeek-and-Suricata log entries

```
{
  "ts": 1052146262.937361,
  "uid": "CVKjZo2GrV8DM0Fvo5",
  "id.orig_h": "203.241.248.20",
  "id.orig_p": 3051,
  "id.resp_h": "80.4.124.41",
  "id.resp_p": 80,
  "proto": "tcp",
  "service": "http",
  "duration": 6.582984,
  ...
}
```

ARGH!
Community ID

Standardized flow hashing for your NSM tools
future extensibility

ID = version · ‘:’ · base64(sha1(seed · 5-tuple))

basic hashing
logically separate deployments
src/dst IP/port, transport proto
visual compression
Typical Zeek-and-Suricata log entries

```json
{
  "ts":1052146262.937361,
  "uid":"CVKjZo2GrV8DM0Fvo5",
  "id.orig_h":"203.241.248.20",
  "id.orig_p":3051,
  "id.resp_h":"80.4.124.41",
  "id.resp_p":80,
  "proto":"tcp",
  "service":"http",
  "duration":6.582984,
  ...
}

{
  "timestamp": "2003-05-05T07:51...",
  "flow_id": 23963675020689,
  "pcap_cnt": 10,
  "event_type": "alert",
  "src_ip": "203.241.248.20",
  "src_port": 3051,
  "dest_ip": "80.4.124.41",
  "dest_port": 80,
  "proto": "TCP",
  "alert": {
    "action": "allowed",
    "gid": 1,
    "signature_id": 9999999,
    "rev": 1,
    "signature": "PWNED!",
    ...
  }
}
```
Typical Zeek-and-Suricata log entries

```json
{
  "ts": 1052146262.937361,
  "uid": "CVKjZo2GrV8DM0Fvo5",
  "id.orig_h": "203.241.248.20",
  "id.orig_p": 3051,
  "id.resp_h": "80.4.124.41",
  "id.resp_p": 80,
  "proto": "tcp",
  "service": "http",
  "duration": 6.582984,
  ...
}
```

```json
{
  "timestamp": "2003-05-05T07:51...",
  "flow_id": 23963675020689,
  "pcap_cnt": 10,
  "event_type": "alert",
  "src_ip": "203.241.248.20",
  "src_port": 3051,
  "dest_ip": "80.4.124.41",
  "dest_port": 80,
  "proto": "TCP",
  "community_id": "1:ZEYOYMeyZNQC9DAg5BZCtiTKw=",
  "alert": {
    "action": "allowed",
    "gid": 1,
    "signature_id": 99999999,
    "rev": 1,
    "signature": "PWNED!"
  }
}
```
Typical Zeek-and-Suricata log entries

```json
{
  "ts":1052146262.937361,
  "uid":"CVKjZo2GrV8DM0Fvo5",
  "id.orig_h":"203.241.248.20",
  "id.orig_p":3051,
  "id.resp_h":"80.4.124.41",
  "id.resp_p":80,
  "proto":"tcp",
  "service":"http",
  "duration":6.582984,
  ...
  "community_id":
    "1:ZEYOYMeyZNQC9DAdgsBZCtiTKqw="
}
```
Example use cases

- ID correlation across multiple monitor log streams
  - Suricata vs Zeek/Snort/etc
  - Also across installations of the same system
- Associating unidirectional flows in asymmetric routes
  - Via Vlad Grigorescu @ ESnet — thx!
- Standardized data enrichment in SIEM
Current status

- Spec and Python reference implementation:
  https://github.com/corelight/community-id-spec

- Included in Suricata 4.1:
  https://github.com/victorjulien/suricata/tree/feature/flow-community-id/v17

- Zeek package:
  https://github.com/corelight/bro-community-id
# Extensible Event Format (nicknamed EVE) event log in JSON format

- eve-log:

  # Community Flow ID
  # Adds a 'community_id' field to EVE records. These are meant to give
  # records a predictable flow id that can be used to match records to
  # output of other tools such as Bro.
  #
  # Takes a 'seed' that needs to be same across sensors and tools
  # to make the id less predictable.

  # enable/disable the community id feature.
  community-id: true

  # Seed value for the ID output. Valid values are 0-65535.
  community-id-seed: 0
But but but but ...
NO GOD PLEASE
NOOOOOOOOO
Yes, this isn’t feature-complete

- “v1” explicitly targets ease of implementation
  - Get people to run it and provide feedback
- Possible version-2 addition: configuration
  - Example: include VLAN, Q-in-Q, others (“vlan”)
  - Example: other hashing algorithm (“sha256”)
  - Example: no base encoding (“nobase”)

2:v6n:f34de81f113ae0bd97242a18d1b82ddea1ef9fd4
Performance vs. collisions

- Again, SHA-1 was our choice because of easy availability
- There are many other (non-secure) hash functions
  - Murmur2, djb2, …
- We’ll see!
Factoring in time

- Currently not included
- Obvious use of time blocks induces risk of ID divergence
  - E.g. rounding to nearest day means a change just before and after midnight — multiple monitors may not be so closely synced
- Clever ideas for “time-approximate fuzzy hashing” welcome!
Other base encodings

- Zeek uses base62 for its ID strings
  Ea6PGGTh0j801GYQNSkx1l3Az6C
- Bitcoin uses base58
  2Raiz2fPQxk3E4hERp1zStCMaN8b
- Length vs. parse-/readability vs. performance tradeoff
Anonymity vs. reversibility

- Hashing means the ID is not reversible
- This enables certain use cases
  - “Have you seen this flow?”
- But reversibility may be desirable
  - Sure, you could simply write it out …
    127.0.0.1/80/192.168.12.234/3434/tcp
- Spectrum with a lot of design points
Could come with a library

- Small base set of languages
- Would simplify implementation in other monitors
Thanks!

- Remember: intentionally basic v1
- Feedback very welcome
  - Github tickets or Suricata & Zeek mailing lists
- Spec and Python reference implementation:
  https://github.com/corelight/community-id-spec