ntopng and Suricata:
Merging Network Visibility and Security

Luca Deri <deri@ntop.org>, @lucaderi
Alfredo Cardigiano <cardigliano@ntop.org>
About ntop

- ntop develops open source network traffic monitoring applications. All code is available at https://github.com/ntop

- ntop is a community: http://t.me/ntop_community

- Part of the Intel Innovator program.

- ntop is also the name of the first app we released in 1998, a web-based network monitoring application (today ntopng).

- Today our tools range from traffic monitoring (ntopng, nProbe), high-speed packet capture (PF_RING), Deep-Packet Inspection (nDPI), traffic recording (n2disk), DDoS mitigation (nScrub), IDS/IPS acceleration.
Network Visibility

- Network visibility ensures that you are able to see everything happening on a network. It includes:
  - Network performance
  - Application performance
  - Devices discovery

- ntopng is a web-based open-source traffic analysis application that aims to provide full network visibility.
Uncorrelated Security Events

- Suricata, as well as other IDS systems, is commonly used to generate alarms when security threats are detected, and produce logs with suspicious network activities.

- There are many tools collecting logs produced by Suricata, and pushing them to system like ElasticSearch. The best they can do is index them and produce statistics: “Tell me how many Policy Violations we got today”.

- Threat detection is typically limited to a single session (see decode-events.c, app-layer-events.c) and it is (mostly) based on signatures matching. Suricata is basically a pure network sensor with no mechanisms for correlating information across multiple flows or hosts.
Augmented Security

• Network administrators need a clear picture of the traffic flowing into their network and place security events in the right context.

• Correlating security events with network traffic provides a better visibility of what’s going on and the root cause of threats.

• Single events that can be considered harmless when looking at them individually, could be small pieces of bigger harmful events.
ntopng Troubleshooting [1/2]
# ntopng Troubleshooting [2/2]

<table>
<thead>
<tr>
<th>Date/Time</th>
<th>Duration</th>
<th>Severity</th>
<th>Alert Type</th>
<th>Drilldown</th>
<th>Description</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>07:31:02</td>
<td>02:32:50</td>
<td>Warning</td>
<td>Ghost Network Detected</td>
<td></td>
<td>Subnet 217.29.66.0/23 does not belong to the [0/0/0/0].</td>
<td>Disable Release</td>
</tr>
<tr>
<td>07:31:02</td>
<td>02:32:50</td>
<td>Error</td>
<td>TCP SYN Scan</td>
<td>Host is under SYN Scan [308 &gt; 30 SYN received]</td>
<td>Disable Release</td>
<td></td>
</tr>
<tr>
<td>07:31:02</td>
<td>02:32:50</td>
<td>Error</td>
<td>TCP SYN Scan</td>
<td>Host is under SYN Scan [127 &gt; 30 SYN received]</td>
<td>Disable Release</td>
<td></td>
</tr>
<tr>
<td>07:31:02</td>
<td>02:32:50</td>
<td>Error</td>
<td>TCP SYN Scan</td>
<td>Host is under SYN Scan [67 &gt; 30 SYN received]</td>
<td>Disable Release</td>
<td></td>
</tr>
<tr>
<td>07:31:02</td>
<td>02:32:50</td>
<td>Error</td>
<td>TCP SYN Scan</td>
<td>Host is under SYN Scan [908 &gt; 30 SYN received]</td>
<td>Disable Release</td>
<td></td>
</tr>
<tr>
<td>07:31:02</td>
<td>02:32:50</td>
<td>Error</td>
<td>TCP SYN Scan</td>
<td>Host is under SYN Scan [905 &gt; 30 SYN received]</td>
<td>Disable Release</td>
<td></td>
</tr>
<tr>
<td>07:31:02</td>
<td>02:32:50</td>
<td>Error</td>
<td>TCP SYN Scan</td>
<td>Host is under SYN Scan [44 &gt; 30 SYN received]</td>
<td>Disable Release</td>
<td></td>
</tr>
<tr>
<td>07:31:02</td>
<td>02:32:50</td>
<td>Error</td>
<td>TCP SYN Scan</td>
<td>Host is a SYN Scan attacker [363 &gt; 50 SYN sent]</td>
<td>Disable Release</td>
<td></td>
</tr>
<tr>
<td>07:31:02</td>
<td>02:32:50</td>
<td>Error</td>
<td>TCP SYN Scan</td>
<td>Host is a SYN Scan attacker [42 &gt; 30 SYN received]</td>
<td>Disable Release</td>
<td></td>
</tr>
<tr>
<td>07:31:02</td>
<td>02:32:50</td>
<td>Error</td>
<td>TCP SYN Scan</td>
<td>Host is a SYN Scan attacker [168 &gt; 50 SYN sent]</td>
<td>Disable Release</td>
<td></td>
</tr>
</tbody>
</table>

Showing 1 to 10 of 233 rows
ntopng Features and Limitations [1/2]

- Host system and containers monitoring through eBPF
- Process, container, POD and user statistics

Full path: useful for drill-down in case of security alerts
ntopng Features and Limitations [2/2]

- ntopng features:
  - Network traffic metrics
  - Anomaly detection
  - Blacklists for malware detection

- It lacks security features including:
  - Threat detection
  - Signatures support
  - File extraction
Suricata Limitations

- It does not use any DPI (Deep Packet Inspection) techniques to identify traffic regardless of the port it uses:

- Running a service on a non standard port might be invisible to it.

- No information about flows/protocols not dissected by Suricata.

- No encrypted traffic analysis (i.e. Cisco Joy-like technologies) beside protocol fingerprinting: the idea is to be able to decode traffic, but unencrypted traffic is becoming rare, and this has impact on visibility.

- It does not provide any facility that could help users to understand the “big picture” (e.g. ARP scan, DNS negative/positive response ratio, or too many host active flows with respect) as it focuses on per-flow analysis.

```bash
alert tcp $HOME_NET any -> $EXTERNAL_NET ![25,587,6666:7000,8076] (msg:"ET POLICY IRC Channel JOIN on non-standard port"
```
Motivation: Unify Visibility and Security [1/2]

- Suricata is a great tool for dissecting selected protocols, extracting key metrics, and emitting alerts based on flow content driven by external rules.

- ntopng is able to collect information from various sources (packets, NetFlow, sFlow), analyse them in a comprehensive format, and emit alerts. All in one place, with minimal requirements.

- What if we can unify these two open source tools into a single tool able to provide the best solution for complementing security and visibility? Seamlessly.
Motivation: Unify Visibility and Security [2/2]

- Benefits for the Suricata community:
  - Provide a web GUI to Suricata. Someone might say: there are many (ELK-based) tools that do that. True but they lack network visibility, require third parties DBs/tools, and are not been designed for networking/security.
  - Enhance Suricata with network metrics not reported by the tool.
  - Provide existing Suricata users with ntop features (e.g. nIndex-based efficient flow-storage or Slack-based alerts).

- Benefits for the ntop community:
  - Add the benefits of signature-based traffic analysis.
  - Merge Suricata traffic alerts with those already handled by ntopng to implement the best of both worlds.
ntopng Architecture

C++ Core Engine

Packet Processing  Flow Collection

{ REST:API }

Lua

influxdb

nDPI

redis

ØMQ
Suricata Eve

- The Suricata EVE output facility outputs events in JSON format.
- Events include:
  - Flow records (à la Netflow)
  - Alerts (signature matches)
  - Application layer metadata (HTTP, DNS, TLS, …)
  - Extracted files information
Syslog Collector Interface

- Ntopng implements Syslog-over-TCP ingestion to collect Syslog records from remote clients.

- Syslog records are processed by Lua modules based on the source application.
Alerts Ingestion

- Alerts generated by Suricata are collected through a Syslog interface.
- Binding the Syslog interface to a physical interface in ntopng we are able to:
  - Correlate events coming from Suricata with traffic processed by ntopng.
  - See network metrics and alerts (as well as other information coming from Suricata) in the same logical interface.
Configuration

- `ntopng -i eth0 -i syslog://127.0.0.1:9999`

*User’s Guide at https://www.ntop.org/guides/ntopng*
Suricata (Syslog) Interface
# Flow Alerts

![Flow Alerts](image)

## Flow Alerts Table

<table>
<thead>
<tr>
<th>Date/Time</th>
<th>Duration</th>
<th>Count</th>
<th>Severity</th>
<th>Alert Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00:17 ago</td>
<td>-</td>
<td>1</td>
<td>Error</td>
<td>External Alert</td>
<td>Detected TROJAN alert: W32/AlerSpy RAT Checkin [Emerging Threats] [Flow: 00:50:56:3B:68:87] [1064 [00:50:56:2F:7A:09]]</td>
</tr>
<tr>
<td>03:07 ago</td>
<td>-</td>
<td>1</td>
<td>Warning</td>
<td>External Alert</td>
<td>Detected HTTP alert: unable to match response to request [Suricata] [Flow: 9C:15:C2:B7:72:3E]</td>
</tr>
</tbody>
</table>

Showing 1 to 3 of 3 rows
### Flow Details

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol / Application</td>
<td>TCP / Unknown (Unspecified)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>First / Last Seen</td>
<td>27/10/2019 14:57:15 [01:17 ago]</td>
<td>27/10/2019 14:58:05 [00:27 ago]</td>
<td></td>
</tr>
<tr>
<td>Total Traffic</td>
<td>Total: 2.39 KB →</td>
<td>Goodput: 1.81 KB (62.7 %) →</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Client → Server: 10 Pkts / 2.30 KB →</td>
<td>Client → Server: 10 Pkts / 600.00 Bytes →</td>
<td></td>
</tr>
</tbody>
</table>

### Round-Trip Time Breakdown

- **Client → Server Estimated Distance**: 20.278 Km, 12.581 Miles
- **Application Latency**: 100.956 ms
- **Packet Inter-Arrival Time [Min / Avg / Max]**:
  - Client → Server: 100 ms / 227.5 ms / 381 ms
  - Client → Server: 100 ms / 227.5 ms / 381 ms
- **TCP Packet Analysis**:
  - Lost: 1 Pkts / 1 Pkts
- **Max (Estimated) TCP Throughput**:
  - Client → Server: 5.09 Mbit/s
  - Client → Server: 5.09 Mbit/s
- **TCP Flags**:
  - Client → Server, SYN, PUSH, ACK
  - Client → Server, SYN, PUSH, ACK

#### Flow Alerted
- Detected TROJAN alert: W32/AlertSpy RAT Checkin [Emerging Threats]

#### Additional Flow Elements
- **Community ID**: 1:F8YukX3ZExaMV913x5y420
- **Suricata Flow ID**: 64789950018662
L7 Metadata Ingestion

- Application layer metadata for selected protocols (e.g. HTTP, DNS, TLS, …) are generated by Suricata and collected through the Syslog interface.

- The Suricata protocol parser and stream reassembly engine can also be used to extract and store files to disk (e.g. from HTTP, SMTP, FTP, …).

- All metadata are ingested by ntopng and are used to compute metrics and run analysis (those natively supported) or just listed as “Additional Information”.
## HTTP & File Info

<table>
<thead>
<tr>
<th>Flow Reset by the client.</th>
</tr>
</thead>
</table>
| Actual / Peak Throughput  | 0 bit/s → / 0 bit/s  
| HTTP                      | GET  
| Server Name               | www.repstatic.it  
| URL                       | www.repstatic.it/minify/sites/repubblica/video/config.mti_db.ca...  
| Response Code             | 200  

### Additional Flow Elements

| File Gaps | No  
|-----------|-----  
| File Name | /content/nazionale/img/2016/02/21/162944540-83640f59-a5f15-4b7e-b06a-cc859d376af7-th.jpg  
| File Size | 8768  
| File State | CLOSED  
| File Stored | No  
| HTTP Content Length | 8768  
| HTTP Mime Type | image/jpeg  
| HTTP Protocol | HTTP/1.1  
| HTTP Referer | http://www.repubblica.it/sport/2016/02/21/italia/83640f59-a5f15-4b7e-b06a-cc859d376af7-th.jpg  
| Suricata Application Protocol | http  
| Suricata Flow ID | 569590231274712  

---

github.com/ntop
Flow Records Ingestion

- Suricata as a NetFlow-like flow exporter.
- Flow information generated by Suricata are collected through a Syslog interface, together with alerts.
- In this working mode, ntopng collects flows instead of processing packet.
- Drawback: ntopng cannot compute most of the Network metrics as it does not have packets visibility.
## Flows List

### Recently Active Flows

<table>
<thead>
<tr>
<th>Application</th>
<th>Protocol</th>
<th>Client</th>
<th>Server</th>
<th>Duration</th>
<th>Breakdown</th>
<th>Actual Thpt</th>
<th>Total Bytes</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLS</td>
<td>TCP</td>
<td>MyMacbook</td>
<td>configuration.apple.com</td>
<td>&lt; 1 sec</td>
<td>Server</td>
<td>0 bps</td>
<td>0 Bytes</td>
<td>configuration.apple.com</td>
</tr>
<tr>
<td>TLS</td>
<td>TCP</td>
<td>MyMacbook</td>
<td>configuration.apple.com</td>
<td>&lt; 1 sec</td>
<td>Server</td>
<td>0 bps</td>
<td>0 Bytes</td>
<td>configuration.apple.com</td>
</tr>
<tr>
<td>TLS</td>
<td>TCP</td>
<td>MyMacbook</td>
<td>configuration.apple.com</td>
<td>&lt; 1 sec</td>
<td>Server</td>
<td>0 bps</td>
<td>0 Bytes</td>
<td>configuration.apple.com</td>
</tr>
<tr>
<td>TLS</td>
<td>TCP</td>
<td>MyMacbook</td>
<td>configuration.apple.com</td>
<td>&lt; 1 sec</td>
<td>Server</td>
<td>0 bps</td>
<td>0 Bytes</td>
<td>configuration.apple.com</td>
</tr>
<tr>
<td>TLS</td>
<td>TCP</td>
<td>MyMacbook</td>
<td>configuration.apple.com</td>
<td>&lt; 1 sec</td>
<td>Server</td>
<td>0 bps</td>
<td>0 Bytes</td>
<td>configuration.apple.com</td>
</tr>
<tr>
<td>TLS</td>
<td>TCP</td>
<td>MyMacbook</td>
<td>t.paypal.com</td>
<td>&lt; 1 sec</td>
<td>Server</td>
<td>0 bps</td>
<td>0 Bytes</td>
<td>t.paypal.com</td>
</tr>
<tr>
<td>TLS</td>
<td>TCP</td>
<td>MyMacbook</td>
<td>gspb1-ssl.is.apple.com</td>
<td>&lt; 1 sec</td>
<td>Server</td>
<td>0 bps</td>
<td>0 Bytes</td>
<td>gspb1-ssl.is.apple.com</td>
</tr>
<tr>
<td>HTTP</td>
<td>TCP</td>
<td>MyMacbook</td>
<td><a href="http://www.repstatic.it">www.repstatic.it</a></td>
<td>&lt; 1 sec</td>
<td>Server</td>
<td>0 bit/s</td>
<td>0 Bytes</td>
<td><a href="http://www.repstatic.it/class/im">www.repstatic.it/class/im</a>...</td>
</tr>
</tbody>
</table>
Flow Details w/o Packets
Flow Details w/ Packets
Ongoing Activities

Analysis and Consolidation

Containers

Feeds and Blacklists

eBPF

Packet Processing

NetFlow

IPFIX

sFlow

Data

Alerts

Work In Progress

github.com/ntop
Final Remarks

- Network security and visibility is now possible.
- Comprehensive merge of Suricata alerting information with ntopng traffic analysis.
- Benefits for the whole open source community, as well the ntopng and Suricata communities.
- Hopefully closer integration using nDPI into Suricata for characterising traffic unknown to Suricata.