Rapid Prototyping of Flow-Based Detection Methods Using Complex Event Processing

Petr Velan, Martin Husák, Daniel Tovarňák
Institute of Computer Science, Masaryk University
Brno, Czech Republic
{velan|husakm|tovarnak}@ics.muni.cz

Abstract

Detection of network attacks is the first step to network security. Many different methods for attack detection were proposed in the past. However, descriptions of these methods are often not complete and it is difficult to verify that the actual implementation matches the description. In this demo paper, we propose to use Complex Event Processing (CEP) for developing detection methods based on network flows. By writing the detection methods in an Event Processing Language (EPL), we can address the above-mentioned problems. The SQL-like syntax of most EPLs is easily readable so the detection method is self-documented. Moreover, it is directly executable in the CEP system, which eliminates inconsistencies between documentation and implementation. The demo will show a running example of a multi-stage HTTP brute force attack detection using Esper and its EPL.

Multi-Stage Attacks

In a sample attack considered in this demo, an attacker tries to take over a a content management system (CMS). First, the attacks scans the network for running web-servers. Then, the attacker checks presence of a CMS such as WordPress or Joomla by requesting URLs typical for the CMS, such as login page. Finally, attacker performs brute-force password attack on CMS login page to get access.

Sample multi-stage attack:

1. **Network scan:** TCP SYN scan on port 80 on all hosts in the network.
2. **HTTP scan:** requesting */wp-login.php* from all active web servers.
3. **Brute-force password attack:** numerous requests for */wp-login.php* on a webserver where such URL is present.

Flow Processing Tools

Flow Probe:

- Captures packets from network
- Tracks uni- or bi-directional connections
- Aggregates connection information
- Exports aggregated flow records

Flow Collector:

- Captures flow records from probes
- Transforms data: anonymization, normalization, format conversion
- Stores or sends data for further processing

HTTP Brute-Force Detection

We include a sample of a query that can be used to detect HTTP brute-forcing. The example goes as follows:

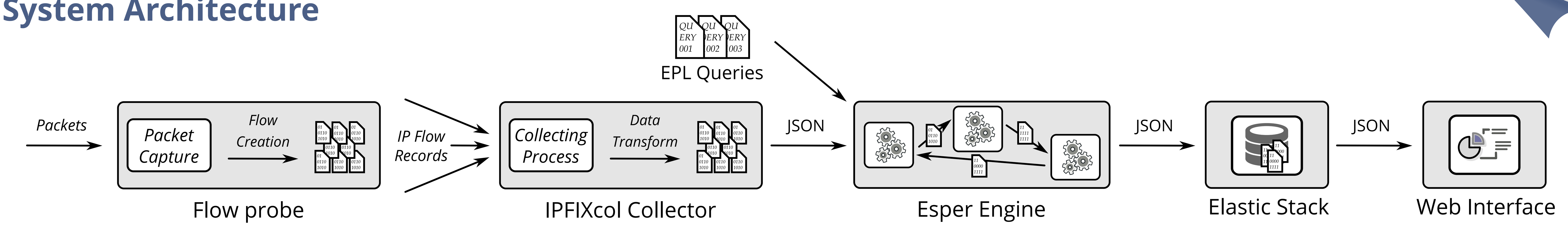
```
@Name('BruteForce')
SELECT
  ipfix.sourceIPv4Address as Attacker,
  ipfix.destinationIPv4Address as Destination,
  ipfix.HTTPRequestHost as Host,
  ipfix.HTTPRequestURL as URL,
  count(ipfix.sourceIPv4Address) as AtkCount
FROM IPFIX.win:time(1 hour)
WHERE
  ipfix.HTTPRequestURL LIKE '%login%'
  or
  ipfix.HTTPRequestURL LIKE '%admin%'
GROUP BY
  ipfix.sourceIPv4Address,
  ipfix.destinationIPv4Address,
  ipfix.HTTPRequestURL
HAVING count(ipfix.sourceIPv4Address) > 50;
```

Correlation of Method's Outputs

This query example illustrates a correlation of outputs of individual detection methods:

```
@Name('Output')
SELECT
  TCPSYNscan.attacker as attacker,
  TCPSYNscan.atkCount as TCPSYNscanCount,
  HTTPscan.atkCount as HTTPscanCount,
  BruteForce.atkCount as BruteForceCount
FROM
  TCPSYNscan.win:time(5 hours),
  HTTPscan.win:time(5 hours),
  BruteForce.win:time(5 hours)
WHERE
  TCPSYNscan.attacker = HTTPscan.attacker
  AND
  TCPSYNscan.attacker = BruteForce.attacker;
```

System Architecture



Acknowledgement

This research was supported by the Security Research Programme of the Czech Republic 2015 - 2020 (BV III / 1 VS) granted by the Ministry of the Interior of the Czech Republic under No. VI20162019029 The Sharing and analysis of security events in the Czech Republic.

https://csirt.muni.cz/
@csirtmu
https://sabu.cesnet.cz/
@CESNET_CERTS

Sources:
https://github.com/CSIRT-MU/FlowCEP

Esper and Event Processing Language

- Fast (> 6 M events per second per CPU)
- Scalable (horizontal scale-out, balancing)
- Embeddable (Java and .NET), standalone platform
- Low Latency (in the range of microseconds)
- SQL-Standard Compliant
- Open Source

CSIRT-MU