SPONSORED BY THE
Federal Ministry of Education and Research

Visualization of Network Security Policy Evaluation

Bastian Hellmann Marcel Reichenbach Leonard Renners Volker Ahlers
University of Applied Sciences and Arts Hannover, Germany

Motivation

- Components that use policies to detect attacks and unwanted behaviour in a network do not share their state
- This makes the analysis of why a specific evaluation returns wrong or unexpected results hard.
- Combining both the sensor data as well as the configuration (policy) of a network security tool and its evaluation can help.

The Interface for Metadata Access Points (IF-MAP)

- Specification by the Trusted Computing Group for standardized metadata exchange
- Data model: undirected graph with Identifiers, Metadata and Links
- Communication model: publish/subscribe system, multiple clients and one server
- Data can be adapted to arbitrary domains (in this case: network security)

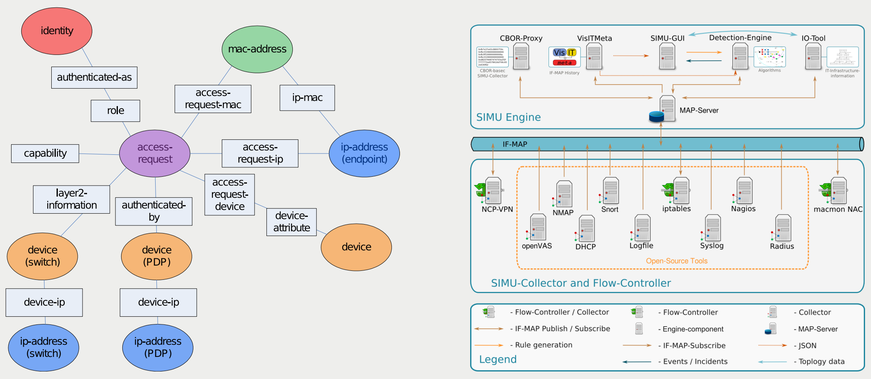

Figure: IF-MAP example graph and exemplary architecture (from the SIMU project)

VisITMeta: Visualising IF-MAP Graphs

- Open-source tool that visualizes IF-MAP graphs
- Result of a 3-year research project in Germany
- Main features: (a) Persistence of history, (b) calculation and visualization of graph deltas, (c) filters and searches and (d) highlighting changes

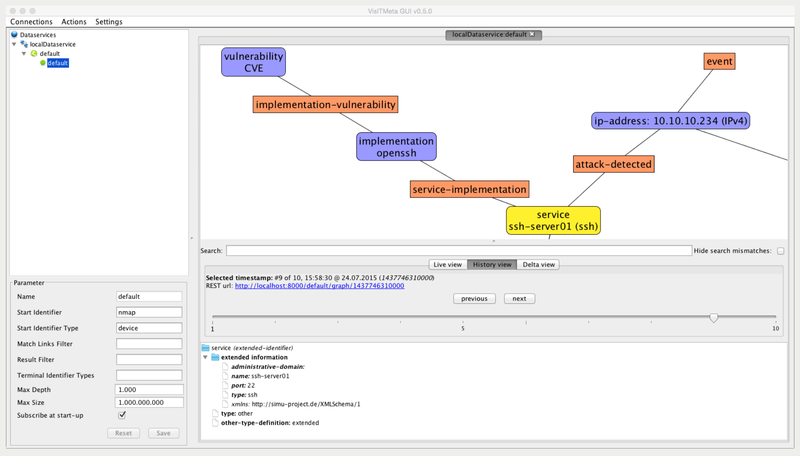

Figure: Screenshot of *VisITMeta* GUI (v0.5.0)

irondetect: An exemplary policy-based detection engine

- Open-source policy-based tool that performs signature checks and anomaly detection on IF-MAP graphs
- Interchangeable methods to detect anomalies
- Considers the context of data (time, location, ...)

```
...

anomaly {
    anoHighTrafficSmartphone := trafficHintSmartphone > 0.5;
}

signature {
    sigCamera := "smartphone.sensor.cameraisused" = "true"
        ctxWorkingHours;
    sigPortOpen :=
        count ("vulnerability-scan-result.vulnerability.port") > "0"
        ctxTrustedInfrastructureComponent;
}

condition {
    conDataLeakDetected := anoHighTrafficSmartphone
        and sigCamera
        and sigPortOpen;
}

action {
    alert := "alert.name" "Data leakage detected";
}

rule {
    dataLeakage := if conDataLeakDetected do alert;
}

...
```

irondetect - Signatures

#	Device	ID	Value	Timestamp
29	af0f	sigPortOpen	☑	2012-12-13T09:37:02+01:00
28	af0f	sigCamera	☑	2012-12-13T09:37:02+01:00
27	af0f	sigSuspiciousApp	☑	2012-12-13T09:37:02+01:00
26	af0f	sigPortOpen	☑	2012-12-13T09:36:53+01:00
25	af0f	sigCamera	☑	2012-12-13T09:36:53+01:00
24	af0f	sigSuspiciousApp	☑	2012-12-13T09:36:53+01:00
23	af0f	sigPortOpen	☑	2012-12-13T09:36:42+01:00
22	af0f	sigCamera	☑	2012-12-13T09:36:42+01:00
21	af0f	sigSuspiciousApp	☑	2012-12-13T09:36:42+01:00
20	af0f	sigPortOpen	☑	2012-12-13T09:36:32+01:00
19	af0f	sigCamera	☑	2012-12-13T09:36:32+01:00
18	af0f	sigSuspiciousApp	☑	2012-12-13T09:36:32+01:00
17	af0f	sigPortOpen	☑	2012-12-13T09:36:29+01:00
16	af0f	sigCamera	☑	2012-12-13T09:36:29+01:00
15	af0f	sigSuspiciousApp	☑	2012-12-13T09:36:29+01:00
14	af0f	sigPortOpen	☑	2012-12-13T09:36:21+01:00
13	af0f	sigCamera	☑	2012-12-13T09:36:21+01:00
12	af0f	sigSuspiciousApp	☑	2012-12-13T09:36:21+01:00
11	af0f	sigPortOpen	☑	2012-12-13T09:36:11+01:00
10	af0f	sigCamera	☑	2012-12-13T09:36:11+01:00
9	af0f	sigSuspiciousApp	☑	2012-12-13T09:36:10+01:00
8	af0f	sigPortOpen	☑	2012-12-13T09:36:06+01:00
7	af0f	sigCamera	☑	2012-12-13T09:36:06+01:00
6	af0f	sigSuspiciousApp	☑	2012-12-13T09:36:06+01:00
5	af0f	sigPortOpen	☐	2012-12-13T09:35:56+01:00
4	af0f	sigCamera	☑	2012-12-13T09:35:56+01:00
3	af0f	sigSuspiciousApp	☑	2012-12-13T09:35:56+01:00
2	af0f	sigCamera	☐	2012-12-13T09:35:36+01:00
1	af0f	sigSuspiciousApp	☑	2012-12-13T09:35:36+01:00

Figure: (Part of) Example policy file of *irondetect* and Screenshot of *irondetect* GUI

Policy Visualization

- Mapping the policy model of *irondetect* to IF-MAP entities.

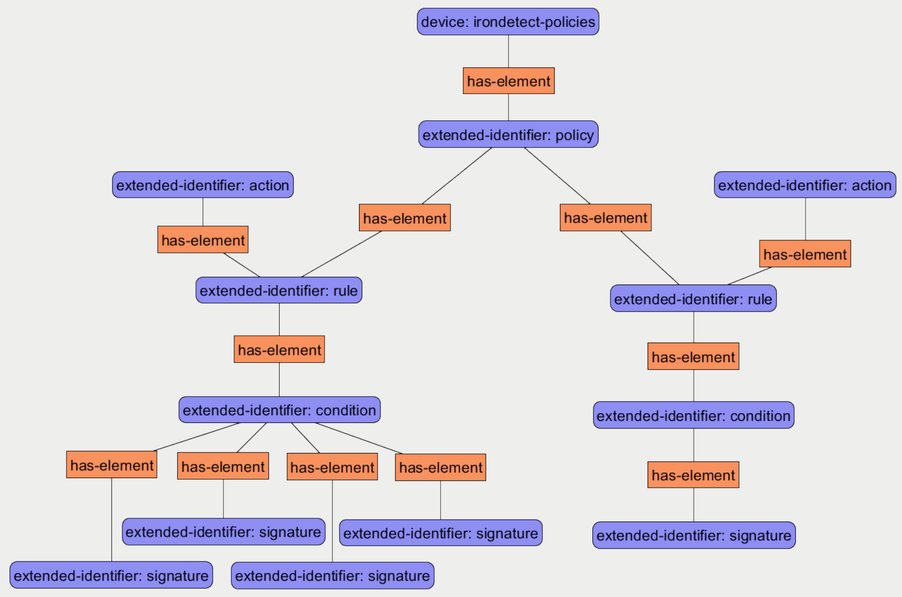

Figure: An *irondetect* policy transformed to IF-MAP

Visualizing the Results of Policy Evaluation

- Publish the inner state of the evaluation to IF-MAP.
- Connections between policy elements and the responsible sensor data that induced a concrete result are made visible to the user.

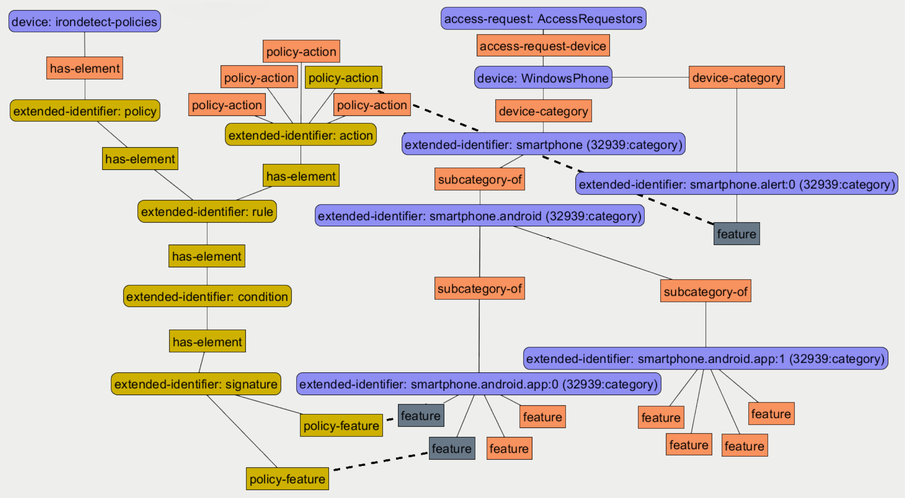

Figure: Results of policy evaluation by *irondetect*

Analysis of Policy Evaluation

- Visualizing both sensor data and policy data allows analysis of different situations:
  - Detect misconfiguration
  - Understand false positives
  - Trace back evaluations

Current Progess and Future Work

- *irondetect* and *VisITMeta* are available as open-source software via GitHub.
- The code to visualize the policy and its evaluation for both *irondetect* and *VisITMeta* components will be released within the next few months.
- Policy modification via the GUI (e.g. as a consequence of too many false positives).
- Adapt additional analysis components to also publish their policies and evaluation.

Contact information

- Email: trust@f4-i.fh-hannover.de
- Website: `http://trust.f4.hs-hannover.de/`
- GitHub: `https://github.com/trustathsh`