Mind your Ps and Qs:

Performing crypto sanity checks with D4.

Team CIRCL
https://www.d4-project.org/

November 12, 2019

Jean-Louis Huynen

D4 - Problem statement

- CSIRTs (or private organisations) build their **own honeypot, honeynet or blackhole monitoring network**
- Designing, managing and operating such infrastructure is a tedious and resource intensive task
- **Automatic sharing** between monitoring networks from different organisations is missing
- Sensors and processing are often seen as blackbox or difficult to audit

D4 - Objective

- Based on our experience with MISP[1] where sharing played an important role, we transpose the model in D4 project
- Keeping the protocol and code base **simple and minimal**
- Allowing every organisation to **control and audit their own sensor network**
- Extending D4 or **encapsulating legacy monitoring protocols** must be as simple as possible
- Ensuring that the sensor server has **no control on the sensor** (unidirectional streaming)
- Don't force users to use dedicated sensors and allow **flexibility of sensor support** (software, hardware, virtual)

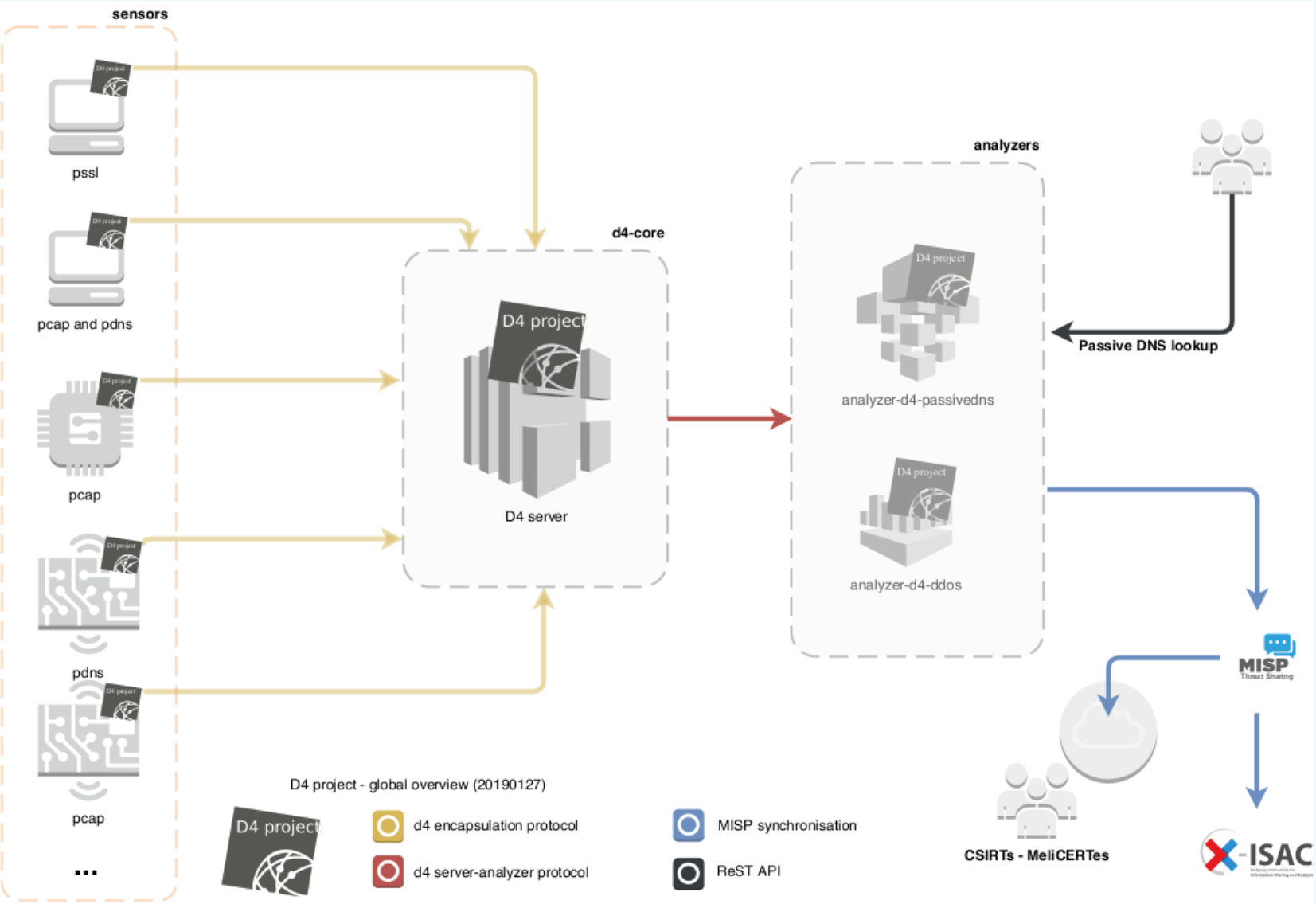
[1] https://github.com/MISP/MISP

D4 - (short) History

- D4 Project (co-funded under INEA CEF EU program) started - **1st November 2018**
- D4 encapsulation protocol version 1 published - **1st December 2018**
- v0.1 release of the D4 core[2] including a server and simple D4 C client - **21st January 2019**
- First version of a golang D4 client[3] running on ARM, MIPS, PPC and x86 - **14th February 2019**

[2] https://www.github.com/D4-project/d4-core
[3] https://www.github.com/D4-project/d4-goclient/

D4 - Overview


sensors
pssl
pcap and pdns
pcap
pdns
pcap
...
d4-core
D4 project
D4 server
analyzers
D4 project
analyzer-d4-passivedns
D4 project
analyzer-d4-ddos
Passive DNS lookup
MISP
Threat Sharing
CSIRTs - MeliCERTes
X-ISAC
D4 project - global overview (20190127)
D4 project
d4 encapsulation protocol
d4 server-analyzer protocol
MISP synchronisation
ReST API

Snake Oil Crypto - Problem Statement

IoT devices **are often the weakest devices** on a network:

- Usually the result of cheap engineering,
- sloppy patching cycles,
- sometimes forgotten–not monitored,
- few hardening features enabled,

We feel a bit safer when they use TLS, but should we?

Snake Oil Crypto - TLS Fingerprinting

Keep a log of links between:

- x509 certificates,
- ports,
- IP address,
- client (ja3),
- server (ja3s),

  "JA3 is a method for creating SSL/TLS client fingerprints that should be easy to produce on any platform and can be easily shared for threat intelligence."[4]

Pivot on additional data points during Incident Response

[4] https://github.com/salesforce/ja3

Snake Oil Crypto - Objectives

Collect and **store** x509 certificates and TLS sessions:

- Public keys type and size,
- moduli and public exponents,
- curves parameters.

Detect anti patterns in crypto:

- Moduli that share one prime factor,
- Moduli that share both prime factors, or private exponents,
- Small factors,
- Nonces reuse / common preffix or suffix, etc.

Snake Oil Crypto - RSA on IoT

Researchers have shown that several devices generated their public keys at boot time without enough entropy[5]:

```
prng.seed(seed)
p = prng.generate_random_prime()
// prng.add_entropy()
q = prng.generate_random_prime()
n = p*q
```

Given n=pq and n' = pq' it is trivial to recover the shared p by computing their Greatest Common Divisor (GCD), and therefore both private keys[6].

[5] Bernstein, Heninger, and Lange: http://facthacks.cr.yp.to/
[6] http://www.loyalty.org/~schoen/rsa/

Snake Oil Crypto - GCD

In Snake-Oil-Crypto we compute GCD[7] between:

- between certificates having the same issuer,
- between certificates having the same subject,
- on keys from various sources (PassiveSSL, Certificate Transparency, shodan, censys, etc.),

“Check all the keys that we know of for vendor X”

[7] using Bernstein’s Batch GCD algorithm

Snake Oil Crypto - MISP feed

2019-11-08	**Name:** crypto-material **References:** 0 **Referenced by:** 6 **uses Object 13800 (network: x509)** **uses Object 13801 (network: x509)** **uses Object 13802 (network: x509)** **uses Object 13803 (network: x509)** **uses Object 13804 (network: x509)** **uses Object 13805 (network: x509)**		
2019-11-08	Other	**p:** text	12732045980491482532629620809854872609730718866846479950748763 99251101386987265586481573653124576541684265313376164608426942 41928677042183313561239786148 69
2019-11-08	Other	**q:** text	None
2019-11-08	Other	**rsa-modulus-size:** text	1024
2019-11-08	Other	**type:** text	RSA

Snake Oil Crypto - MISP feed

The MISP feed

- **Allows** for checking automatic checking by an IDS on hashed values,
- **contains** thousands on broken keys from a dozen of vendors,
- **will be accessible upon request (info@circl.lu).**

In the future:

- **Automatic** the vendor checks by performing TF-IDF on x509's subjects,
- **automatic** vendors notification.

First release

- ✓ sensor-d4-tls-fingerprinting [8]: **Extracts** and **fingerprints** certificates, and **computes** TLSH fuzzy hash.
- ✓ analyzer-d4-passivessl [9]: **Stores** Certificates / PK details in a PostgreSQL DB.
- snake-oil-crypto [10]: **Performs** crypto checks, push results in MISP for notification
- lookup-d4-passivessl [11]: **Exposes** the DB through a public REST API.

[8] `github.com/D4-project/sensor-d4-tls-fingerprinting`
[9] `github.com/D4-project/analyzer-d4-passivessl`
[10] `github.com/D4-project/snake-oil-crypto`
[11] `github.com/D4-project/lookup-d4-passivessl`

Use it

- **Manage** your own sensors and servers, **find** shameful bugs and **fill** in github issues
- Even better, **send** Pull Requests!
- **Share** data to public servers to improve the datasets (and detection, response, etc.)
- **Feed** your MISP instances with D4's findings - **Share** yours
- **Leech** data, **write** your own analyzers, **do** research

Get in touch if you want to join the project, host a sensor or contribute

- Collaboration can include research partnership, sharing of collected streams or improving the software.
- Contact: info@circl.lu
- `https://github.com/D4-Project`
- `https://twitter.com/d4_project`
- `https://d4-project.org`
  - Passive DNS tutorial
  - Data sharing tutorial