D4 Project
Open and collaborative network monitoring

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

2019/07/03
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
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 Overview - Connecting Sensor Networks

- Sensors
- tcpdump
- D4 server
- tcpdump
- D4 core
- D4 project
- analyzer-d4
- d4-client
- ORG A
- ORG B

Protocols:
- d4 encapsulation protocol
- d4 server-analyzer protocol
- ReST API
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/
<table>
<thead>
<tr>
<th>Release</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>analyzer-d4-passivedns-v0.1</td>
<td>Apr. 5, 2019</td>
</tr>
<tr>
<td>analyzer-d4-passivessl-0.1</td>
<td>Apr. 25, 2019</td>
</tr>
<tr>
<td>analyzer-d4-pibs-v0.1</td>
<td>Apr. 8, 2019</td>
</tr>
<tr>
<td>BGP-Ranking-1.0</td>
<td>Apr. 25, 2019</td>
</tr>
<tr>
<td>d4-core-v0.1</td>
<td>Jan. 25, 2019</td>
</tr>
<tr>
<td>d4-core-v0.2</td>
<td>Feb. 14, 2019</td>
</tr>
<tr>
<td>d4-core-v0.3</td>
<td>Apr. 8, 2019</td>
</tr>
<tr>
<td>d4-goclient-v0.1</td>
<td>Feb. 14, 2019</td>
</tr>
<tr>
<td>d4-goclient-v0.2</td>
<td>Apr. 8, 2019</td>
</tr>
<tr>
<td>d4-server-packer-0.1</td>
<td>Apr. 25, 2019</td>
</tr>
<tr>
<td>IPASN-History-1.0</td>
<td>Apr. 25, 2019</td>
</tr>
<tr>
<td>sensor-d4-tls-fingerprinting-v0.1</td>
<td>Apr. 25, 2019</td>
</tr>
</tbody>
</table>

see https://github.com/D4-Project
CIRCL will host a server instance for organisations willing to contribute to a public dataset without running their own D4 server:

- Blackhole DDoS
- Passive DNS
- Passive SSL
- Gene⁴ / WHIDS⁵ (sysmon)
- BGP mapping
- egress filtering mapping
- Radio-Spectrum monitoring: 802.11, BLE, GSM, etc.

---

⁴https://github.com/oxrawsec/gene
⁵https://github.com/oxrawsec/whids
D4 encapsulation protocol version 1

version (8) - Version of the header

*type* (8) - Data encapsulated type

*uuid* (128) - Sensor UUID

*timestamp* (64) - Encapsulation time

*hmac* (256) - Header authentication (HMAC-SHA256-128)

*size* (32) - Payload size

---

Stream of information (text or binary)

---

https://www.d4-project.org
<table>
<thead>
<tr>
<th>Name</th>
<th>bit size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>version</td>
<td>uint 8</td>
<td>Version of the header</td>
</tr>
<tr>
<td>type</td>
<td>uint 8</td>
<td>Data encapsulated type</td>
</tr>
<tr>
<td>uuid</td>
<td>uint 128</td>
<td>Sensor UUID</td>
</tr>
<tr>
<td>timestamp</td>
<td>uint 64</td>
<td>Encapsulation time</td>
</tr>
<tr>
<td>hmac</td>
<td>uint 256</td>
<td>Authentication header (HMAC-SHA-256-128)</td>
</tr>
<tr>
<td>size</td>
<td>uint 32</td>
<td>Payload size</td>
</tr>
<tr>
<td>Type</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>pcap (libpcap 2.4)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>meta header (JSON)</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>generic log line</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>dnscap output</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>pcapng (diagnostic)</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>generic NDJSON or JSON Lines</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>generic YAF (Yet Another Flowmeter)</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>passivedns CSV stream</td>
<td></td>
</tr>
<tr>
<td>254</td>
<td>type defined by meta header (type 2)</td>
<td></td>
</tr>
</tbody>
</table>
D4 header includes an easy way to extend the protocol (via type 2) without altering the format. Within a D4 session, the initial D4 packet(s) type 2 defines the custom headers and then the following packets with type 254 is the custom data encapsulated.

```json
{
    "type": "ja3-jl",
    "encoding": "utf-8",
    "tags": [
        "tlp:white"
    ],
    "misp:org": "5b642239-4db4-4580-adf4-4ebd950d210f"
}
```
D4 core server\(^6\) is a complete server to handle clients (sensors) including the decapsulation of the D4 protocol, control of sensor registrations, management of decoding protocols and dispatching to adequate decoders/analysers.

D4 server is written in Python 3.6 and runs on standard GNU/Linux distribution.

\(^6\)https://github.com/D4-project/d4-core
**D4 server handling**

D4 server reconstructs the encapsulated stream from the D4 sensor and saves it in a Redis stream.

- Support TLS connection
- Unpack D4 header
- Verify client secret key (HMAC)
- Check blocklist
- Filter by types (Only accept one connection by type-UUID - except: type 254)
- Discard incorrect data
- Save data in a Redis Stream (unique for each session)
The D4 server provides a **web interface** to manage D4 sensors, sessions and analyzer.

- Get Sensors status, errors and statistics
- Get all connected sensors
- Manage Sensors (stream size limit, secret key, ...)
- Manage Accepted types
- UUID/IP blocklist
- Create Analyzer Queues
## D4 SERVER - MAIN INTERFACE

<table>
<thead>
<tr>
<th>UUID</th>
<th>Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>4019794</td>
<td>c0bb49e788964718af4dde4c0ab89bc</td>
</tr>
<tr>
<td>47820</td>
<td>bbccf7a43aed47aa844badc50262f5aba</td>
</tr>
<tr>
<td>27183</td>
<td>37d2f040fc074a9ab2c8af49059b679325</td>
</tr>
<tr>
<td>8401</td>
<td>1b06b4ab8a754e9ae3d4d073b38f0e5</td>
</tr>
<tr>
<td>1022</td>
<td>de1df62d862b494a830f1f78ec27fca5</td>
</tr>
<tr>
<td>4046981</td>
<td>1: pcap (libpcap 2.4)</td>
</tr>
<tr>
<td>57243</td>
<td>8: passivedns CSV stream</td>
</tr>
</tbody>
</table>

2019/05/20
D4 SERVER - SERVER MANAGEMENT
## D4 Server - Sensor Overview

### Sensor Details

<table>
<thead>
<tr>
<th>UDI</th>
<th>Last Seen</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>one.osf/eight.osf</td>
<td>2019-05-30 11:38:53</td>
<td>OK Connected</td>
</tr>
<tr>
<td>three.osf/six.osf</td>
<td>2019-05-30 11:38:53</td>
<td>OK Connected</td>
</tr>
<tr>
<td>seven.osf/eight.osf</td>
<td>2019-05-30 11:38:53</td>
<td>OK Connected</td>
</tr>
<tr>
<td>five.osf/six.osf</td>
<td>2019-05-30 11:38:53</td>
<td>OK Connected</td>
</tr>
<tr>
<td>four.osf/six.osf</td>
<td>2019-05-30 11:38:53</td>
<td>OK Connected</td>
</tr>
<tr>
<td>two.osf/eight.osf</td>
<td>2019-05-30 11:38:53</td>
<td>OK Connected</td>
</tr>
<tr>
<td>one.osf/seven.osf</td>
<td>2019-05-30 11:38:53</td>
<td>OK Connected</td>
</tr>
<tr>
<td>three.osf/five.osf</td>
<td>2019-05-30 11:38:53</td>
<td>OK Connected</td>
</tr>
<tr>
<td>seven.osf/five.osf</td>
<td>2019-05-30 11:38:53</td>
<td>OK Connected</td>
</tr>
<tr>
<td>five.osf/four.osf</td>
<td>2019-05-30 11:38:53</td>
<td>OK Connected</td>
</tr>
<tr>
<td>four.osf/two.osf</td>
<td>2019-05-30 11:38:53</td>
<td>OK Connected</td>
</tr>
<tr>
<td>two.osf/one.osf</td>
<td>2019-05-30 11:38:53</td>
<td>OK Connected</td>
</tr>
</tbody>
</table>
A distributed Network telescope to observe DDoS attacks
DDoS Attacks produce an observable side-effect:

<table>
<thead>
<tr>
<th>Number of packets</th>
<th>Date (month / day)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$3 \times 10^6$</td>
<td>01/10</td>
</tr>
<tr>
<td>$2.5 \times 10^6$</td>
<td>01/24</td>
</tr>
<tr>
<td>$2 \times 10^6$</td>
<td>02/07</td>
</tr>
<tr>
<td>$1.5 \times 10^6$</td>
<td>02/21</td>
</tr>
<tr>
<td>$1 \times 10^6$</td>
<td>03/07</td>
</tr>
<tr>
<td>$500000$</td>
<td></td>
</tr>
</tbody>
</table>

![Graph of backscatter traffic volume per 5 minutes in 2019](https://www.circl.lu/)

- Backscatter traffic
- TCP traffic

https://www.circl.lu/
What can be derived from backscatter traffic?

- **External point of view on ongoing Denial of Service attacks:**
  - **Confirm** if there is a DDoS attack
  - **Recover** time line of attacked targets
  - **Confirm** which services (DNS, webserver, ...)  
  - **Observe** Infrastructure changes

- **Assess the state of an infrastructure under denial of service attack**
  - **Detect** failure/addition of intermediate network equipments, firewalls, proxy servers etc
  - **Detect** DDoS mitigation devices

- **Create** models of DoS/DDoS attacks
D4 IN THIS SETTING

D4 - for data collection and processing:
- **provide** various points of observation in non contiguous address space,
- **aggregate** and **mix** backscatter traffic collected from D4 sensors,
- **perform** analysis on big amount of data.

D4 - from a end-user perspective:
- **provide** backscatter analysis results,
- **provide** daily updates,
- **provide** additional relevant (or pivotal) information (DNS, BGP, etc.),
- **provide** an API and search capabilities.
✓ analyzer-d4-pibs, an analyzer for a D4 network sensor:

- **processes** data produced by D4 sensors (pcaps),
- **displays** potential backscatter traffic on standard output,
- **focuses** on TCP SYN flood in this first release.

[7]https://github.com/D4-project/analyzer-d4-pibs
- CIRCL (and other CSIRTs) have their own passive DNS\(^8\) collection mechanisms
- Current **collection models** are affected with DoH\(^9\) and centralised DNS services
- DNS answers collection is a tedious process
- **Sharing Passive DNS stream** between organisation is challenging due to privacy

\(^8\)https://www.circl.lu/services/passive-dns/
\(^9\)DNS over HTTPS
Potential Strategy

- Improve **Passive DNS collection diversity** by being closer to the source and limit impact of DoH (e.g. at the OS resolver level)
- Increasing diversity and **mixing models** before sharing/storing Passive DNS records
- Simplify process and tools to install for **Passive DNS collection by relying on D4 sensors** instead of custom mechanisms
- Provide a distributed infrastructure for mixing streams and filtering out the sharing to the validated partners
 analyzer-d4-passivedns\textsuperscript{10}, an analyzer for a D4 network sensor:
  \begin{itemize}
    \item \textbf{processes} data produced by D4 sensors (in passivedns CSV format\textsuperscript{11}),
    \item \textbf{ingests} these into a \textbf{Passive DNS server} which can be queried later to search for the Passive DNS records,
    \item \textbf{provides} a lookup server (using on redis-compatible backend) that is a Passive DNS REST server compliant to the Common Output Format\textsuperscript{12}.
  \end{itemize}

\textsuperscript{10}https://github.com/D4-project/analyzer-d4-passivedns
\textsuperscript{11}https://github.com/gamelinux/passivedns
\textsuperscript{12}https://tools.ietf.org/html/draft-dulaunoy-dnsop-passive-dns-cof-04
Passive SSL revamping
Objectives - 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.”\(^{13}\)

**Pivot** on additional data points during Incident Response

\(^{13}\)https://github.com/salesforce/ja3
Collect and store x509 certificates and TLS sessions:
- Public keys type and size,
- moduli and exponents,
- curves parameters.

Detect anti patterns in crypto:
- Shared Public Keys,
- Moduli that share one prime factor,
- Moduli that share both prime factor,
- Small factors,
- Nonces reuse / common prefix or suffix, etc.
sensor-d4-tls-fingerprinting: Extracts and fingerprints certificates, and computes TLSH fuzzy hash.

analyzer-d4-passivessl: Stores Certificates / PK details in a PostgreSQL DB.

lookup-d4-passivessl: Exposes the DB through a public REST API.
- **Sensitive information sanitization** by specialized analyzers
- **Previewing datasets** collected in D4 sensor network and providing **open data stream** (if contributor agrees to share under specific conditions)
- **Leverage MISP sharing communities** to augment Threat Intelligence, and provide accurate metrology.
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