Improving Passive DNS collection

with D4 Project

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

2019/03/29

Alexandre Dulaunoy

Problem statement

- CIRCL (and other CSIRTs) have their own passive DNS[1] collection mechanisms
- Current **collection models** are affected with DoH[2] and centralised DNS services
- DNS answers collection is a tedious process
- **Sharing Passive DNS stream** between organisation is challenging due to privacy

[1] `https://www.circl.lu/services/passive-dns/`

[2] 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

First release

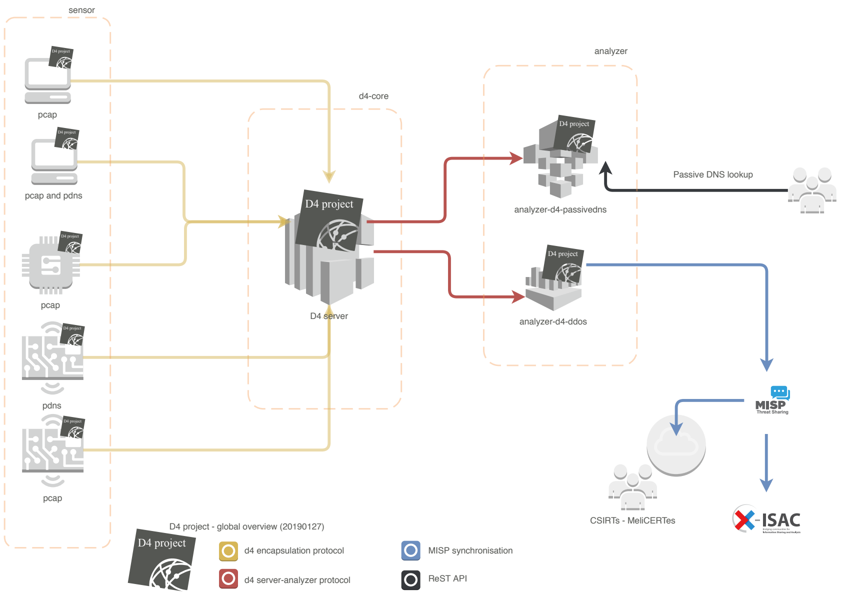
- analyzer-d4-passivedns[3] is an analyzer for a D4 network sensor. The analyser can process data produced by D4 sensors (in passivedns CSV format[4]
- Ingest these into a Passive DNS server which can be queried later to search for the Passive DNS records
- The lookup server is a Passive DNS ReST server compliant to the Common Output Format[5]

[3] https://github.com/D4-project/analyzer-d4-passivedns

[4] https://github.com/gamelinux/passivedns

[5] https://tools.ietf.org/html/draft-dulaunoy-dnsop-passive-dns-cof-04

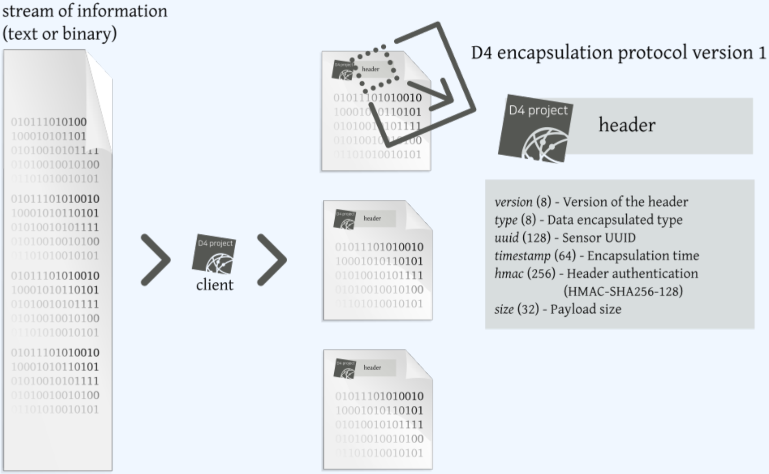
D4 Overview


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

Roadmap (next 2 months)

- Passive DNS analyzer (alpha version released)
- Passive SSL collector and analyzer
- Backscatter DDoS traffic analyzer
- **Default server** (blackhole monitoring or Passive DNS collector) at CIRCL for organisations willing to contribute without running their own D4 server

D4 encapsulation protocol



stream of information
(text or binary)

client

D4 encapsulation protocol version 1

header

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

https://www.d4-project.org

D4 Header

Name	bit size	Description
version	uint 8	Version of the header
type	uint 8	Data encapsulated type
uuid	uint 128	Sensor UUID
timestamp	uint 64	Encapsulation time
hmac	uint 256	Authentication header (HMAC-SHA-256-128)
size	uint 32	Payload size

D4 Header

Type	Description
0	Reserved
1	pcap (libpcap 2.4)
2	meta header (JSON)
3	generic log line
4	dnscap output
5	pcapng (diagnostic)
6	generic NDJSON or JSON Lines
7	generic YAF (Yet Another Flowmeter)
8	passivedns CSV stream
254	type defined by meta header (type 2)

D4 meta header

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

- 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)

D4 server - worker handler

After the stream is processed depending of the type using dedicated worker.

- Worker Manager (one by type)
  - Check if a new session is created and valid data are saved in a Redis stream
  - Launch a new Worker for each session
- Worker
  - Get data from a stream
  - Reconstruct data
  - Save data on disk (with file rotation)
  - Save data in Redis. Create a queue for D4 Analyzer(s)

D4 server - type 254 worker handler

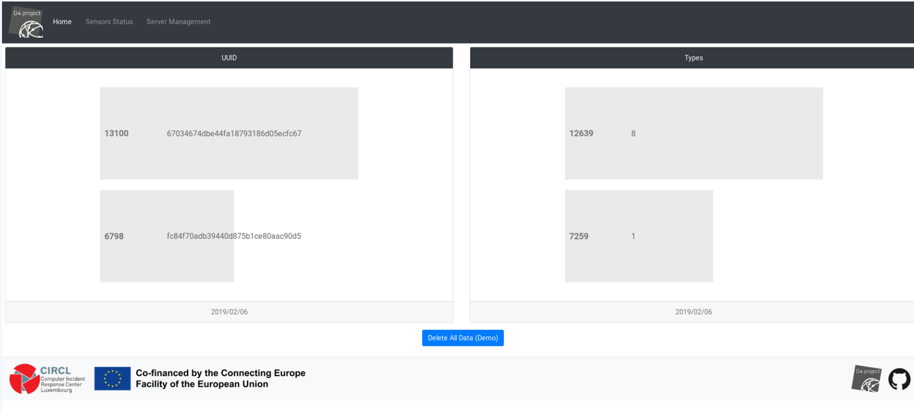
- Worker 2
  - Get type 2 data from a stream
  - Reconstruct Json
  - Extract extended type name
  - Use default type or special extended handler
  - Save Json on disk
  - Get type 254 data from a stream
  - Reconstruct type 254
  - Save data in Redis. Create a queue for D4 Analyzer(s)

D4 server - management interface

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



UUID

13100 67034674dbe44fa18793186d05ecfc67

6798 fc84f70adb39440d875b1ce80aac90d5

2019/02/06

Types

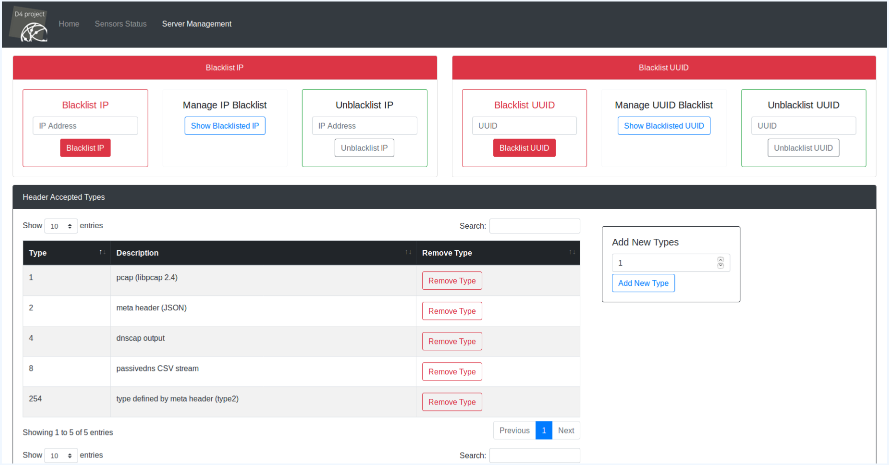
12639 8

7259 1

2019/02/06

Co-financed by the Connecting Europe Facility of the European Union

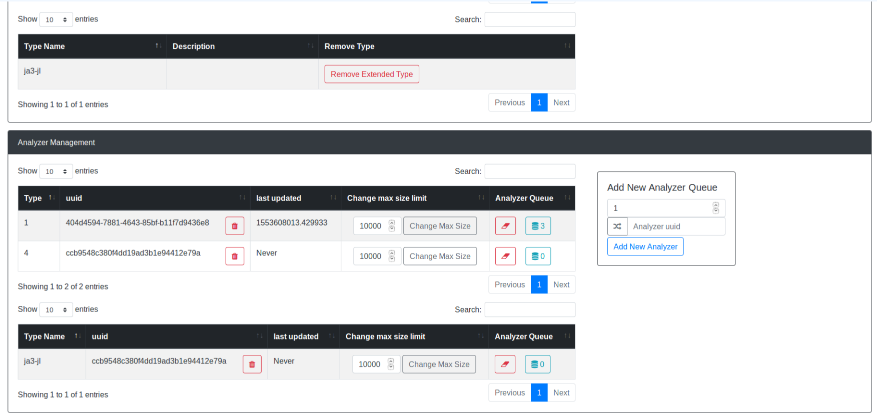
D4 SERVER - SERVER MANAGEMENT



Type	Description	Remove Type
1	pcap (libpcap 2.4)	Remove Type
2	meta header (JSON)	Remove Type
4	dnscap output	Remove Type
8	passivedns CSV stream	Remove Type
254	type defined by meta header (type2)	Remove Type

Showing 1 to 5 of 5 entries

D4 SERVER - SERVER MANAGEMENT



Show 10 entries

Search:

Type Name	Description	Remove Type
ja3-jl		Remove Extended Type

Showing 1 to 1 of 1 entries

Analyzer Management

Show 10 entries

Search:

Type	uuid	last updated	Change max size limit	Analyzer Queue
1	404d4594-7881-4643-85bf-b11f7d9436e8	1553608013.429933	10000 Change Max Size	3
4	ccb9548c380f4dd19ad3b1e94412e79a	Never	10000 Change Max Size	0

Showing 1 to 2 of 2 entries

Show 10 entries

Search:

Type Name	uuid	last updated	Change max size limit	Analyzer Queue
ja3-jl	ccb9548c380f4dd19ad3b1e94412e79a	Never	10000 Change Max Size	0

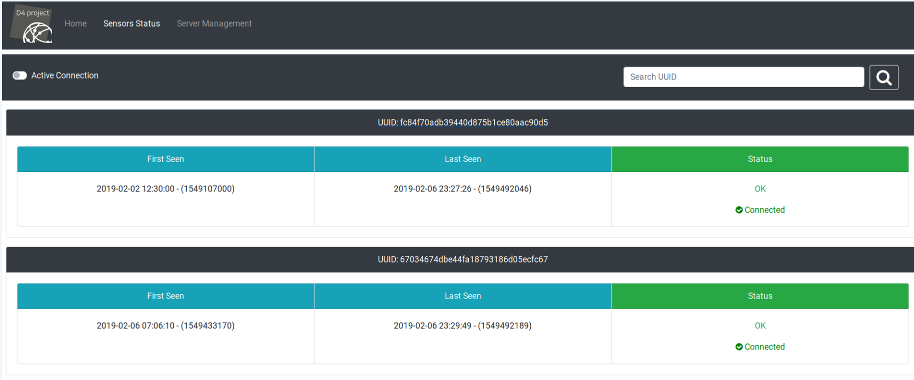
Showing 1 to 1 of 1 entries

Add New Analyzer Queue

1

Analyzer uuid

D4 SERVER - SENSOR OVERVIEW



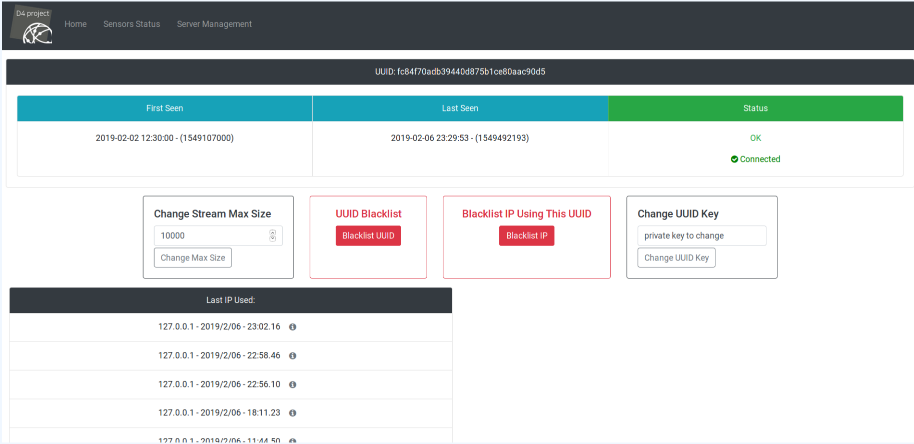
UUID: fc84f70adb39440d875b1ce80aac90d5

First Seen	Last Seen	Status
2019-02-02 12:30:00 - (1549107000)	2019-02-06 23:27:26 - (1549492046)	OK ✓ Connected

UUID: 67034674dbe44fa18793186d05ecfc67

First Seen	Last Seen	Status
2019-02-06 07:06:10 - (1549433170)	2019-02-06 23:29:49 - (1549492189)	OK ✓ Connected

D4 SERVER - SENSOR MANAGEMENT



UUID: fc84f70adb39440d875b1ce80aac90d5

First Seen	Last Seen	Status
2019-02-02 12:30:00 - (1549107000)	2019-02-06 23:29:53 - (1549492193)	OK ✓ Connected

Change Stream Max Size

10000

UUID Blacklist

Blacklist IP Using This UUID

Change UUID Key

private key to change

Last IP Used:
127.0.0.1 - 2019/2/06 - 23:02.16 ℹ
127.0.0.1 - 2019/2/06 - 22:58.46 ℹ
127.0.0.1 - 2019/2/06 - 22:56.10 ℹ
127.0.0.1 - 2019/2/06 - 18:11.23 ℹ
127.0.0.1 - 2019/2/06 - 11:44.50 ℹ

Use-case: migrating a legacy network capture model into a D4 network sensor

Remote network capture

CIRCL operated honeybot for multiple years using a simple model of remote network capture.

Definition (Principle)

- KISS (Keep it simple stupid) - Unix-like
- Linux & OpenBSD operating systems

Sensor

```
tcpdump -l -s 65535 -n -i vr0 -w - ’( not port
   $PORT and not host $HOST )’ | socat - OPENSSL
   -CONNECT:$COLLECTOR:$PORT,cert=/etc/openssl/
   client.pem,cafile=/etc/openssl/ca.crt,verify=1
```

Remote network capture

Limitations

- Scalability → one port per client
- Identification and registration of the client
- Integrity of the data

Encapsulating streams in D4

- Inspired by the unix command `tee`
- Read from standard input
- Add the d4 header
- Write it on standard output

Using D4 native client

```
tcpdump -n -s0 -w - | ./d4 -c ./conf | socat -
   OPENSSL-CONNECT:$D4-SERVER-IP-ADDRESS:$PORT,
   verify=1
```

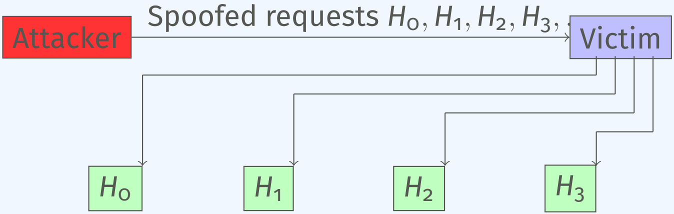
Configuration directory

Parameter	Explanation
type	see D4 Header slide
source	standard input
key	HMAC key
uuid	Identifier of the sensor
version	version of the sensor
destination	standard output
snaplen	length of data being read & written

Use-case: D4 analyzer to detect DDoS attacks in backscatter traffic

OBSERVING SYN FLOODS ATTACKS IN BACKSCATTER TRAFFIC

Attack description



Connections
H_0
H_1
H_2
H_3

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 are a target (DNS, webserver, ...)
- Infrastructure changes or updates
- 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 or services
- Create probabilistic models of denial of service attacks

Confirm if there is/was a DDoS attack

Problem

- Distinguish between compromised infrastructure and backscatter
- Look at TCP flags → filter out single SYN flags
- Focus on ACK, SYN/ACK, ...
- Do not limit to SYN/ACK or ACK → ECE (ECN Echo)[7]

```
tshark -n -r capture-20170916110006.cap.gz -T
   fields -e frame.time_epoch -e ip.src -e tcp.
   flags
1505552542.807286000 x.45.177.71 0x00000010
1505552547.514922000 x.45.177.71 0x00000010
```

[7] https://tools.ietf.org/html/rfc3168

Passive Identification of Backscatter (WiP)

```
./pibs -b -r pcap_file.cap
```

Early version is available of PIBS[8] with a focus on TCP traffic.

Options	Explanations
-r	read pcap file
-b	display IPs under DDoS on standard output

Dependencies
libwiretap-dev
libhiredis-dev
libwsutil-dev

[8] https://github.com/D4-project/analyzer-d4-pibs

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