DOTS telemetry related Hackathon activity report

IETF 106 Hackathon

Kaname Nishizuka/NTT Communications

Hackathon Plan

- Draft
  - https://datatracker.ietf.org/doc/draft-reddy-dots-telemetry/
- ~~Preliminary implementation and PoC of DOTS telemetry~~

- Design Review of DOTS telemetry

Design of DOTS telemetry

Purpose

- Giving a maximum capability of conveying normal/ attack traffic related metrics as hints from a DOTS client to a DOTS server and vice versa.

Timing

- pre-mitigation
- post-mitigation

- pre-mitigation

pre-mitigation

pre-mitigation telemetry

Pre-mitigation resources are not bound to any mitigation request.

URI-Path: “telemetry”

module	explanation
telemetry-config	configuration of telemetry
total-*	baseline/capacity
attack-detail	attack information

set beforehand
(not frequently updated)

update of current status

Proposal

- Separate the URI-path of them into 2
  - update of attack-detail doesn’t always require configuration update

pre-mitigation

DOTS client to server telemetry

URI-Path: “telemetry-config” (proposal)

module	explanation
telemetry-config	configuration of telemetry
total-*	baseline/capacity
~~attack-detail~~	~~attack information~~

PUT: convey the telemetry configuration

GET: retrieve the negotiated configuration

DELETE: delete and set the parameters to default values

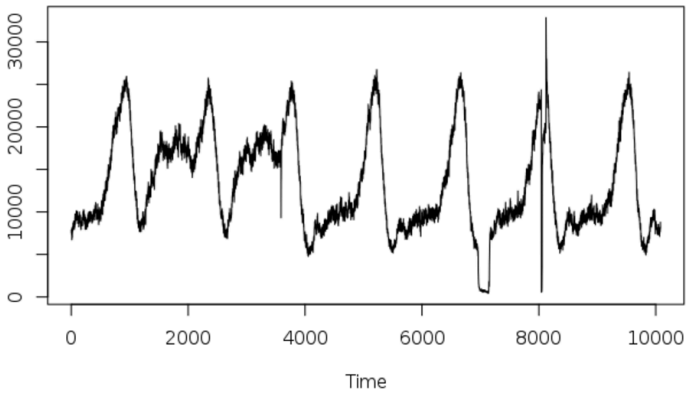
Theoretically it works well with “tcid”(=Telemetry Configuration Identifier)

pre-mitigation

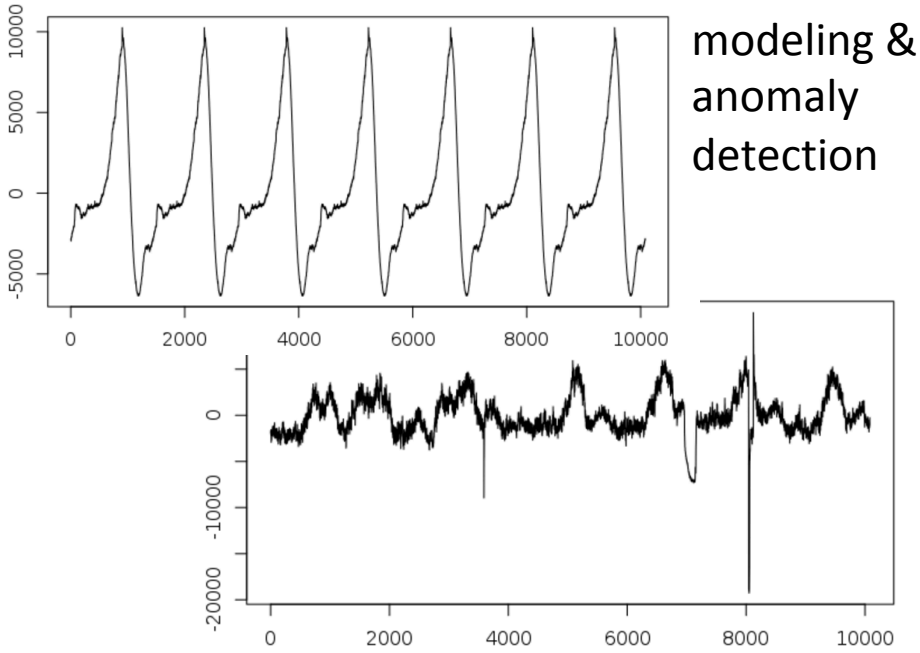
Machine Learning approach consideration

DOTS client

metrics



DOTS server



modeling & anomaly detection

1. Sending “normal traffic baseline” calculated at a DOTS client
2. Sending traffic metrics periodically, then “normal traffic learning” at a DOTS server

ML based approach

pre-mitigation

DOTS client to server telemetry

URI-Path: “pre-mitigation” (proposal)

module	explanation
attack-detail	attack information
normal-traffic	normal traffic related information

PUT: convey the current information of attack/ normal traffic from a DOTS client

GET: retrieve the (historical) traffic information

DELETE: delete all the traffic information

(timestamp will be needed for normal-traffic)

NOTE: If DOTS agents send traffic metrics, it needs to be compared with other approaches like IPFIX

pre-mitigation

DOTS server to client telemetry

URI-Path: “pre-mitigation-attackinfo” (proposal)

module	explanation
attack-detail	attack information
~~normal-traffic~~	~~normal traffic related information~~

- Observe Option set to '0' in the GET request
- receive asynchronous notifications of attack-detail from the DOTS server.

pre-mitigation

Why S-to-C attack info in pre-mitigation stage

Inconsistency in attack knowledge

Scenario

- When a DDoS attack happened, SOC at the DOTS client side can notice something going wrong but cannot figure out which IP address is exactly attacked
- What if SOC at the DOTS server can convey attack-detail to the DOTS client?
- The DOTS client can finally trigger a mitigation request based upon the hint gave via telemetry

- post-mitigation

post-mitigation

post-mitigation telemetry

Post-mitigation resources are bound to existing mitigation-scope.

module	explanation
attack-detail	attack information

Client to Server

- Sent in initial mitigation request(PUT)
- Sent as a part of efficacy update(PUT)

Server to Client

- Sent as a part of mitigation status update

- No new URI-path will be need

- Considerations

percentile calculation

Context of percentile calculation

- period of time (1hour, 1day … 1 month)
- time granularity (1sec, 1min, 5min …)

Inconsistency of them between the DOTS client and server will lead to misunderstanding of attack characteristics

terminology “request”

```
leaf request-ps {
  type uint64;
  description
    "The maximum number of requests allowed per second
     to the target server.";
}
leaf request-client-ps {
  type uint64;
  description
    "The maximum number of requests allowed per second
     to the target server per client.";
}
leaf partial-request-ps {
  type uint64;
  description
    "The maximum number of partial requests allowed per
     second to the target server.";
}
leaf partial-request-client-ps {
  type uint64;
  description
    "The maximum number of partial requests allowed per
     second to the target server per client.";
}
```

Inconsistency of definition of “request” and “partial-request” will also lead to misunderstanding of attack characteristics

(couldn’t find exact definition of what is “request” here)

- Interop status updates

Interop status updates

- Continuous interop testing with Jon (after the last IETF)
- Found several bugs on both sides. There is no significant issue on signal-channel(-38) and data-channel(-31) except one(*).
- Both are willing to test the new DOTS heartbeat spec introduced from v39 draft ASAP
  - will be reported back to WG
- go-dots: https://github.com/nttdots/go-dots
  - kubernetes deployment will be available soon

* usage of RST to cancel Observe requests will not work with DOTS gateway.

Thank You