OpenC2

Protocol: http, mqtt, opendxl, …

Message:
- **Type:** request, reply, notification
- **Headers:** to, from, date-time, request-id, …
- **Body:** content

Content (command):
- **Action:** allow, deny, contain, mitigate …
- **Target**: ip addr, url, file, device, …
- **Args:** …
- **Function:** packet filtering, intrusion detection, endpoint protection, …

```
{
    "action": "deny",
    "target": {
        "url": "http://example.com"
    }
}
```

```
Or using compact JSON:
["deny", {"url": "http://example.com"}]
```

Payload = complete action, response, or notification

Payload

OpenDXL Ontology

Actions:
- Blacklist URL
- Block file by hash
- Quarantine device by MAC addr
- Quarantine device by hostname
- Unblacklist URL
- Unblock file by hash
- Unquarantine device by MAC addr
- Unquarantine device by hostname

Notifications:
- Blacklist URL
- Block file by hash
- Quarantine device by MAC addr
- ...

```
{ "url": "http://example.com" }
```

Payload = arguments

```
{ "status": 200,
  "status_text": "The action succeeded",
  "product": "McAfee TIE" }
```

Response: Documented under Action

OpenDXL Ontology

Quarantine device notifications

Version: `0.0.1`

Quarantine device notifications

Events

/notification/quarantine/device/by_hostname

EVENT

Quarantine device by hostname notification

payload: *object*

EXAMPLE

```
{
    "hostname": "testdevice.local.com"
}
```

PROPERTIES

hostname: *string* required

Internet host name as specified in [RFC1123]

Other Field	Description	Example
product_name	The product that is the source of the message	McAfee Threat Intelligence Exchange (TIE)
product_version	The version of the product that is the source of the message	3.0.0
service_id	The identifier of the DXL service that is the source of the notification	{32cd9168-338f-11e4-0d01-005056946833}

OpenC2

Protocol: http, mqtt, opendxl, …

Message:

- **Type:** request, reply, notification
- **Headers:** to, from, date-time, request id, …
- **Body:** content

Content:

- **Action:** allow, deny, contain, mitigate …
- **Target**: ip addr, url, file, device, …
- **Args:** …
- **Function:** packet filtering, intrusion detection, endpoint protection, …

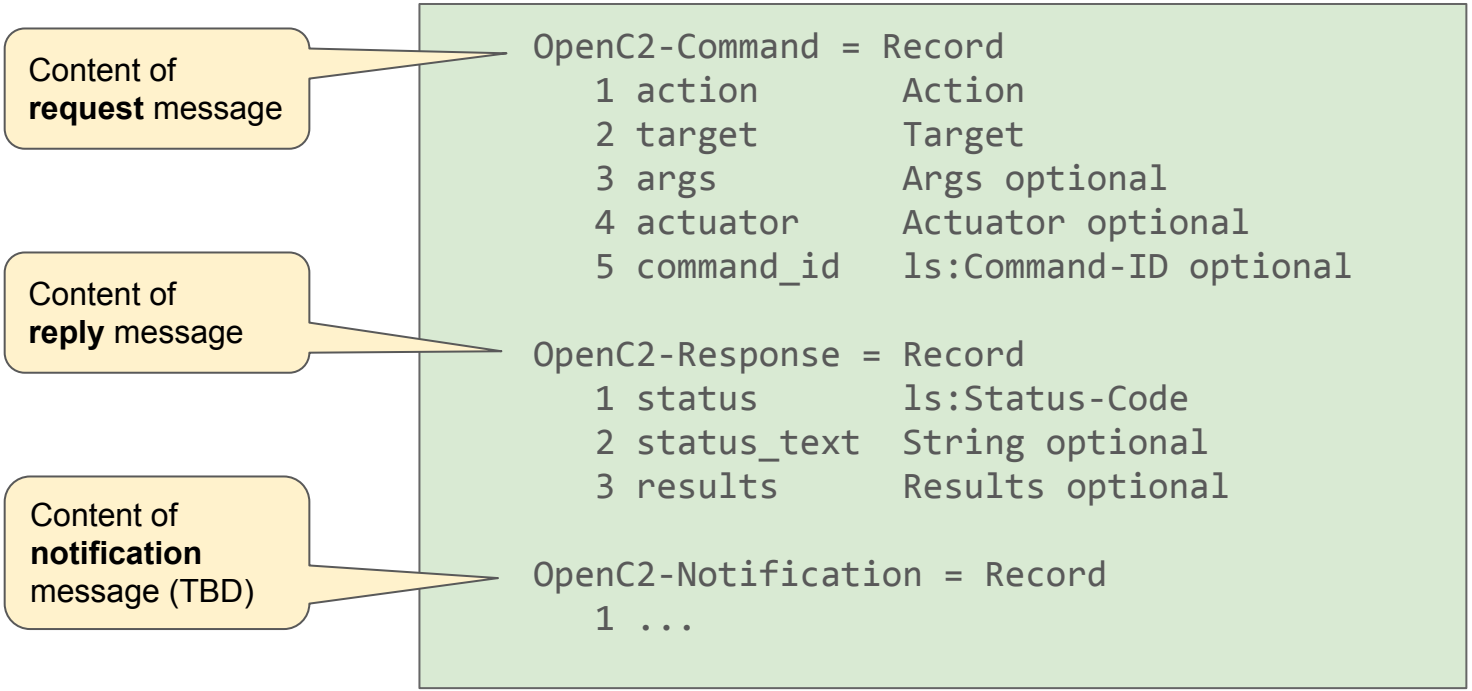
Message

```
{
  "type": "notification",
  "headers": {
    "service_id": "32cd9168-338f-11e4-0d01-0050569
  }
  "body": {
```

Content

```
    "action": "contain",
    "target": {
      "device": {
        "hostname": "testdevice.local.com"
      }
    }
  }
}
```

OpenC2 Content



OpenC2 Functions (Profiles)

packet filtering actions

packet filtering standard targets

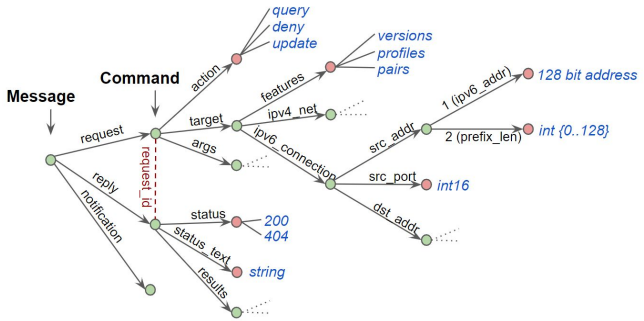
packet filtering extension targets

```
Action = Enumerated
   3 query
   6 deny
   8 allow
  16 update
  20 delete

Target = Choice
   9 features          ls:Features
  10 file              ls:File
  13 ipv4_net          ls:IPv4-Net
  14 ipv6_net          ls:IPv6-Net
  15 ipv4_connection   ls:IPv4-Connection
  16 ipv6_connection   ls:IPv6-Connection
1024 slpf/             slpf:AP-Target

slpf:AP-Target = Choice
   1 rule_number       slpf:Rule-ID
```

packet filtering action-target pairs

```
{
  "query": ["features"],
  "deny": ["ipv4_net", "ipv6_net", ...],
  "allow": ["ipv4_net", "ipv6_net", ...],
  "update": ["file"],
  "delete": ["slpf:rule_number"]
}
```

Design Considerations



OpenC2 is a data API like Falcor and GraphQL

- **"*The data is the API*":** OpenC2 payload contains all information about a request, reply, or notification
- Paths (**/action/quarantine/device/by_hostname)** are titles for documentation but not used in protocol data?
- Scalability: the list of paths (Actions * Targets * Arguments) could become very long

Fundamental Decision:

- Should OpenDXL payload be Message or Content?
- Original HTTP body was Content. Based on implementation experience, http will also support Message body

Questions:

- What Functions (endpoint management, firewalling, packet filtering, …) does OpenDXL Ontology address?
- What is the use case for notifications? /notification/blacklist/url is sent from where to where for what purpose?

Falcor: https://netflix.github.io/falcor/
GraphQL: https://graphql.org/

OpenDXL Ontology Paths	OpenC2 mapping to OpenDXL Paths
/action/blacklist/url	/action/deny/url
/action/block/file/by_hash	/action/deny/file/by_hash
/action/quarantine/device/by_mac_address	/action/contain/device/by_mac_address
/action/quarantine/device/by_hostname	/action/contain/device/by_hostname
/action/unblacklist/url	/action/allow/url
/action/unblock/file/by_hash	/action/allow/file/by_hash
/action/unquarantine/device/by_mac_address	/action/allow/device/by_mac_address
/action/unquarantine/device/by_hostname	/action/allow/device/by_hostname
/notification/blacklist/url	/notification/deny/url
/notification/block/file/by_hash	/notification/deny/file/by_hash
/notification/quarantine/device/by_mac_address	/notification/contain/device/by_mac_address
/notification/quarantine/device/by_hostname	/notification/contain/device/by_hostname
/notification/unblacklist/url	/notification/allow/url
/notification/unblock/file/by_hash	/notification/allow/file/by_hash
/notification/unquarantine/device/by_mac_address	/notification/allow/device/by_mac_address
/notification/unquarantine/device/by_hostname	/notification/allow/device/by_hostname
	/action/query/features
	/action/query/blinky/device
	/action/set/blinky/display
	/action/deny/ipv4_net
	/action/deny/ipv4_net/slpf/direction/slpf/insert_rule
	/action/delete/slpf/rule_number

Observations from mapping process:

- Similar meaning: "blacklist", "block" mapped to **deny**. "quarantine" mapped to **contain**.
- Explicit undo: "blacklist - unblacklist", "block - unblock", "contain - uncontain" - all undo's mapped to **allow**?
  - Asymmetry is disturbing
- Non-obvious undo: OpenC2 slpf: undo both **allow** and **deny** implemented as **delete rule_number**
  - OpenC2 should directly model "ruleset" object?
- OpenDXL Path represents a single leaf value, cannot represent multiple values
  - deny/ipv4_connection/*by_source_addr,* deny/ipv4_connection/*by_source_port*
  - Path should stop at target: /action/quarantine/device, with by_mac or by_hostname or both specified in payload
- OpenDXL Path does not represent an Extension / Function (distributed development of profiles using namespaces)
  - OpenC2 Actuator field is exclusively a filter parameter, should be multivalued and moved into Args
  - "blinky" and "slpf" targets extend the core OpenC2 language - how are extensions represented in Ontology?

Takeaway

Whatever format OpenDXL Ontology specification uses, it can be mapped to and from OpenC2. OpenC2 specification is a Graph information model (vertices + edges).

Goals:

- Define mapping rules to make the correspondence as clear as possible
- Define OpenDXL Ontology formally (as data) to enable automated translation:
  - to OpenDXL Ontology specification (page template with variables)
  - to OpenDXL on-the-wire message content
  - to / from OpenC2 Message and Content data