Timely Incident Response

Vasileios Mavroeidis, Security Researcher
Information and Cyber Security Research Group
University of Oslo, Norway
Email: vasileim@ifi.uio.no

Kamer Vishi, Security Researcher
Information and Cyber Security Research Group
University of Oslo, Norway
Email: kamerv@ifi.uio.no
Cyber Defense of Today

- Defenses are statically configured and operate in isolation

- Cyber Response
  - Slow - Time to implement
  - Manual

On average, it takes 214 days to identify a malicious or criminal attack, and 77 days to contain and recover [Carbon Black]
Defenses are **DYNAMICALLY** configured and are part of an **ORCHESTRATION** process.

**Coordinated Defense (multi-part response actions) in Cyber Relevant Time**

"IOC might take 60 minutes to investigate manually can be researched in 30 seconds with Orchestration" [FireEye]

**How?**

- Need to speak the same language and protocols
- Need to share what we know about attacks in cyber-relevant time (CTI)

**Standardization is a Key Enabler for Automation**
OpenC2 -- Automated Courses of Action?

Open Command and Control (OpenC2) is a concise and extensible language to enable machine to machine communications for purposes of command and control of cyber defence components, subsystems and/or systems in a manner that is agnostic of the underlying products, technologies, transport mechanisms or other aspects of the implementation.

<table>
<thead>
<tr>
<th>Title</th>
<th>Work In Progress (Google Doc)</th>
<th>Latest Published CSD (HTML)</th>
<th>Committee Specification / OASIS Standard</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language Specification</td>
<td>WD09</td>
<td>CSD07 / PDR01 17 Oct 18 PR01 CRM</td>
<td>Future</td>
</tr>
<tr>
<td>Stateless Packet Filtering Actuator Profile</td>
<td>WD05</td>
<td>CSD04 / PRD01 17 Oct 18 PR01 CRM</td>
<td>Future</td>
</tr>
<tr>
<td>HTTPS Transfer Specification</td>
<td>WD04</td>
<td>CSD03 / PRD01 17 Oct 18 PR01 CRM</td>
<td>Future</td>
</tr>
</tbody>
</table>
OpenC2 at a glance

● Unambiguous Machine-to-Machine Communication
● Simplicity
  ○ Low overhead on sensor and actuator
● Focuses on the ‘Acting’ portion of cyber defense
● OpenC2 assumes the following has been done:
  ○ Sensing: *What* triggers the action
  ○ Analytics: *Why*
  ○ Decision: *Which* action
OpenC2 Terminology

- **Actuator**: The device or sensor that executes a native OpenC2 command

- **Orchestrator**: Is a mission manager that will issue the OpenC2 commands to the appropriate actuators, and in the synchronous case, ensure the commands are executed in the correct order

- **Profile**: A minimum to implement set of OpenC2 commands that a class of actuators support

- **OpenC2 Proxy**: Provides a mapping of OpenC2 commands to and from devices that do not natively support OpenC2.
Notional OpenC2 Implementation for Firewalls
CTI + CoA = Automated CoA

CTI + OpenC2 = **Orchestrated** Automated CoA
Notional OpenC2 Implementation
OpenC2 Syntax

- **Action**: The task or activity to be performed
- **Target**: The object of the action
- **Actuator**: The entity that performs the action
- **Arguments and Specifiers**: Additional precision to the commands or the actuators
OpenC2 - Stateless Packet Filtering (SLPF)

- Example: Deny a particular connection
  - Block a particular connection within the domain and do not send a host unreachable

```json
{
    "action": "deny",
    "target": {
        "ip_connection": {
            "protocol": "tcp",
            "src_addr": "192.168.1.1",
            "dst_addr": "81.167.155.132",
            "dst_port": 80
        }
    },
    "args": {
        "start_time": 1534775460000,
        "duration": 500,
        "response_requested": "ack",
        "slpf": {
            "drop_process": "none"
        }
    },
    "actuator": {
        "slpf": {
            "asset_id": "30"
        }
    }
}
```

OpenC2 Command

```
{
    "status": 200
}
```

OpenC2 Response
# OpenC2 Terminology

## Actions Applicable to SLPF

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>query</td>
<td>Initiate a request for information. Used to communicate the supported options and determine the state or settings.</td>
</tr>
<tr>
<td>6</td>
<td>deny</td>
<td>Prevent traffic or access.</td>
</tr>
<tr>
<td>8</td>
<td>allow</td>
<td>Permit traffic or access.</td>
</tr>
<tr>
<td>16</td>
<td>update</td>
<td>Instructs the actuator to update its configuration by retrieving and processing a configuration file and update.</td>
</tr>
<tr>
<td>20</td>
<td>delete</td>
<td>Remove an access rule.</td>
</tr>
</tbody>
</table>

## Targets Applicable to SLPF

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>ip_addr</td>
<td>IP-Addr</td>
<td>The representation of one or more IP addresses (either version 4 or version 6) expressed using CIDR notation.</td>
</tr>
<tr>
<td>15</td>
<td>ip_connection</td>
<td>IP-Connection</td>
<td>A network connection that originates from a source and is addressed to a destination. Source and destination addresses may be either IPv4 or IPv6; both should be the same version</td>
</tr>
<tr>
<td>16</td>
<td>features</td>
<td>Features</td>
<td>A set of items such as action target pairs, profiles versions, options that are supported by the actuator. The target is used with the query action to determine an actuator's capabilities.</td>
</tr>
<tr>
<td>1024</td>
<td>slpf</td>
<td>slpf:Target</td>
<td>Targets defined in the Stateless Packet Filter profile.</td>
</tr>
</tbody>
</table>
Use Case

OpenC2 to Cisco
OpenC2 Schema

\{JSON\} + ASN.1 = JADN

```json
...
{
    "action": "query",
    "target": {
        "openc2": ["schema"]
    }
}
...
```

16
Mapping: OpenC2 - Cisco

**DENY**
```
{
    "action": "deny",
    "target": {
        "ip-connection": {
            "protocol": "tcp",
            "dst_port": "22",
            "dst_addr": "172.20.52.0/24",
            "src_addr": "171.69.198.0/24"
        }
    },
    "actuator": {
        "slpf": {"asset_id":"uio"}
    },
    "args": {
        "command_id": "tekna19_05022019",
        "response_requested": "ack"
    }
}
```

**ALLOW**
```
{
    "action": "allow",
    "target": {
        "ip-connection": {
            "protocol": "tcp",
            "dst_addr": "0.0.0.0/0",
            "src_addr": "0.0.0.0/0"
        }
    },
    "actuator": {
        "slpf": {"asset_id":"uio"}
    },
    "args": {
        "command_id": "tekna20_05022019",
        "response_requested": "ack"
    }
}
```

```
uio(config)# access-list 102 deny tcp 171.69.198.0 0.0.0.255 172.20.52.0 0.0.0.255 eq ssh
uio(config)# access-list 102 permit tcp any any
```
#Validation of the OpenC2 Command

#Action Validation - Needs to be one from the list "query","deny","allow","update","delete"
if (openc2["action"] %in% action_list == FALSE) {print("Action command not in conformance with oc2slpf-v1.0-csprd01")}

#Target Validation - Needs to be one from the list "file","ip_addr","ip_connection","features","slpf"
if (names(openc2$target) %in% target_list == FALSE) {print("Target specifier not in conformance with oc2slpf-v1.0-csprd01")}

#if the target is "ip_connection" we have to validate that one or more of the following elements in the list exist:
#if "src-addr","src-port","dst-addr","dst-port","protocol"
for (i in names(openc2$target)) {
  if (names(openc2$target[i]) == "ip_connection") {
    #target is ip_connection then you check for protocol, src_addr
    for (j in names(openc2$target$ip_connection)) {
      if(j == "protocol"){
        if (openc2$target$ip_connection$protocol %in% l4_protocol_list==FALSE){
          print("Target->ip_connection->protocol specifier not in conformance with oc2slpf-v1.0-csprd01")
        }else print("Target->ip_connection->protocol specifier passed validation")
      }
      else if (j == src_addr) {
        if (is.na(cidr_validation(openc2$target$ip_connection$src_addr))){
          print("Not valid source address, it needs to be of type CIDR")
        }
      }
    }
  }
}
Notional OpenC2 Implementation for Firewalls
We need to block SSH access (remote connection) from any host...

ACTION=DENY,
TARGET=ip_connection
ACTUATOR=slpf
ARGUMENTS=command_id,
response_requested

I see you want me to block SSH access (remote connection) from any host in network..., and send an ack response.
I need to translate this OpenC2 language to CISCO IOS language.

Cisco router! I need you to block a particular service (ssh) for specific hosts...

Got it! Here is your ack receipt.
Thank you for ack receipt.

ACTION=DENY, TARGET=ip_connection
ACTUATOR=slpf
ARGUMENTS=command_id, response_requested
Who is OpenC2?

... and many more!