THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

University of Waikato
Network Research Group

openLI

Configuring a VoIP Intercept

OpenLI Training: Chapter Eleven

Shane Alcock
University of Waikato
New Zealand
shane.alcock@waikato.ac.nz

VoIP Intercepts

- An LEA may request interception of VoIP calls only
  - Equivalent to a phone tap, but for the digital age

- All other data communications are ignored
  - Interception of SIP, RTP, RTMP protocols only
  - May include any video streams as well

SIP

- Session Initiation Protocol
  - Orchestrates and configures a VoIP call between two endpoints

- From SIP traffic, we can derive
  - The identity of the caller
  - The identity of the callee
  - The IP addresses of both participants
  - The UDP ports that will be used for the RTP and RTMP streams

VoIP Identity

- SIP Identity is expressed as <username>@<realm>

- Username
  - May be a typical user string, e.g. `john.smith`
  - May also be a phone number, or a UID number, e.g. `021476611`

- Realm
  - Typically your network domain, e.g. `wand.net.nz`
  - Could also be the SIP server IP, e.g. `192.168.100.27`

VoIP Calls

- Each call has a unique Call-ID
  - Assigned by the participating SIP software
  - This will correspond to a CIN value in your intercepts
    - Same call == same CIN for both IRIs and CCs

VoIP IRIs

- All SIP messages should be intercepted
  - Encoded as IRI records and mediated to the LEA over HI2
  - This includes indirect messages, such as REGISTER

VoIP Intercept Example Scenario

- We operate the network “example.com”
- We run a SIP server at 10.100.50.65:5060
- We have received a VoIP intercept warrant
  - User: 12345678910

Configuring the SIP Server

- First, we need to tell OpenLI about our SIP server
  - Ideally, you would do this *before* you get a warrant!

- POST the SIP server details to the provisioner via REST API
  - http://<PROVIP>:<RESTAPIPORT>/sipserver

Configuring the SIP Server

- The JSON object for a SIP server is very simple

```
{
    “ipaddress”: “10.100.50.65”,
    “port”: “5060”
}
```

Configuring the SIP Server

- Using curl to push the request to the provisioner

```
curl -X POST -H "Content-Type: application/json"
  -d '{
        "ipaddress": "10.100.50.65",
        "port": "5060"
      }'
  http://172.19.0.3:8080/sipserver
```

```
<html><body>OpenLI provisioner configuration was successfully
updated.</body></html>
```

Configuring the SIP Server

- In the collector logs, we should now see

```
openlicollector[12539]: OpenLI: collector has added 10.100.50.65-5060 to its
SIP core server list.
```

Removing the SIP Server

- Using curl to disable a SIP server
  - Note the hyphen in the server identifier!

```
curl -X DELETE http://172.19.0.3:8080/sipserver/10.100.50.65-5060
```

```
<html><body>OpenLI provisioner configuration was successfully
updated.</body></html>
```

SIP Servers

- All traffic for that server + port is designated as SIP traffic
  - Applies to both TCP and UDP
  - SIP content is parsed to derive call identities, RTP streams, etc.

- Absolutely necessary for successful VoIP interception
  - Forgetting to configure SIP servers is a common error

REST API for VoIP Intercepts

- Adding a VoIP intercept via REST
  - POST request
  - Content Type is a JSON object

http://<PROVIP>:<RESTAPIPORT>/voipintercept

REST API for VoIP Intercepts

- The JSON object for our upcoming intercept

```
{
    "liid": "TESTVOIP001",
    "authcc": "NZ",
    "delivcc": "NZ",
    "mediator": 1,
    "agencyid": "mocklea",
    "starttime": 0,
    "endtime": 0,
    "siptargets": [
        { "username": "12345678910" }
    ]
}
```

REST API for VoIP Intercepts

- The JSON object for our upcoming intercept

```
{
    "liid": "TESTVOIP001",
    "authcc": "NZ",
    "delivcc": "NZ",
    "mediator": 1,
    "agencyid": "mocklea",
    "starttime": 0,
    "endtime": 0,
    "siptargets": [
        { "username": "12345678910" }
    ]
}
```

REST API for VoIP Intercepts

- The JSON object for our upcoming intercept

```
{
    "liid": "TESTVOIP001",
    "authcc": "NZ",
    "delivcc": "NZ",
    "mediator": 1,
    "agencyid": "mocklea",
    "starttime": 0,
    "endtime": 0,
    "siptargets": [
        { "username": "12345678910" }
    ]
}
```

REST API for VoIP Intercepts

- The JSON object for our upcoming intercept

```
{
    “liid”: “TESTVOIP001”,
    “authcc”: “NZ”,
    “delivcc”: “NZ”,
    “mediator”: 1,
    “agencyid”: “mocklea”,
    “starttime”: 0,
    “endtime”: 0,
    “siptargets”: [
        { “username”: “12345678910” }
    ]
}
```

REST API for VoIP Intercepts

- The JSON object for our upcoming intercept

```
{
    “liid”: “TESTVOIP001”,
    “authcc”: “NZ”,
    “delivcc”: “NZ”,
    “mediator”: 1,
    “agencyid”: “mocklea”,
    “starttime”: 0,
    “endtime”: 0,
    “siptargets”: [
        { “username”: “12345678910” }
    ]
}
```

REST API for VoIP Intercepts

- The JSON object for our upcoming intercept

```
{
    “liid”: “TESTVOIP001”,
    “authcc”: “NZ”,
    “delivcc”: “NZ”,
    “mediator”: 1,
    “agencyid”: “mocklea”,
    “starttime”: 0,
    “endtime”: 0,
    “siptargets”: [
        { “username”: “12345678910” }
    ]
}
```

REST API for VoIP Intercepts

- The JSON object for our upcoming intercept

```
{
    “liid”: “TESTVOIP001”,
    “authcc”: “NZ”,
    “delivcc”: “NZ”,
    “mediator”: 1,
    “agencyid”: “mocklea”,
    “starttime”: 0,
    “endtime”: 0,
    “siptargets”: [
        { “username”: “12345678910” }
    ]
}
```

REST API for VoIP Intercepts

- Using curl to add the intercept on the provisioner

```
curl -X POST -H "Content-Type: application/json"
  -d '{
        "liid": "TESTVOIP001",
        "authcc": "NZ",
        "delivcc": "NZ",
        "mediator": 1,
        "agencyid": "mocklea",
        "starttime": 0,
        "endtime": 0,
        "siptargets": [
            { "username": "12345678910" }
        ]
    }'
  http://172.19.0.3:8080/voipintercept
```

REST API for VoIP Intercepts

- In the collector logs, we will see success messages

```
openlicollector[12539]: OpenLI: adding new VOIP intercept TESTVOIP001 (start
time 0, end time 0)
openlicollector[12539]: OpenLI: collector received new SIP target
123456789l0@* for LIID TESTVOIP001.
```

REST API for VoIP Intercepts

- There will also be helpful logs on the other components

```
openliprovisioner[2715]: OpenLI provisioner: added new VOIP intercept
TESTVOIP001 via update socket.
```

```
openlimediator[12443]: OpenLI Mediator: received "Activated" HI1
Notification from provisioner for LIID TESTVOIP001 (target agency = mocklea)
```

REST API for VoIP Intercepts

- Modify VoIP intercepts with PUT requests
  - Must include the LIID field in your JSON object
  - If modifying SIP targets, include **ALL** targets you want to keep
  - Other unchanged fields are optional

```
curl -X PUT -H "Content-Type: application/json"
  -d '{
        "liid": "TESTVOIP001",
        "siptargets": [
            { "username": "12345678910" },
            { "username": "jsmith", "realm": "example.com" }
        ]
    }'
  http://172.19.0.3:8080/voipintercept
```

REST API for VoIP Intercepts

- Remove VoIP intercepts with DELETE

```
curl -X DELETE http://172.19.0.3:8080/voipintercept/TESTVOIP001
```

REST API for VoIP Intercepts

- Fetch VoIP intercepts with GET

```
curl -X GET http://172.19.0.3:8080/voipintercept/TESTVOIP001

curl -X GET http://172.19.0.3:8080/voipintercept
```

Next Time

- Let’s intercept a VoIP call!
  - Validate the records that appear at our mock agency