VoIP Security Management

Humberto Abdelnur, Vincent Cridlig, Jérome Bourdellon, Radu State, Olivier Festor

LORIA-INRIA Lorraine, France
{abdelnur, cridligv, bourdellon, state, festor}@loria.fr
http://madynes.loria.fr
Outline

• EnSUITE: Netconf management suite
• VoIP management with Netconf
• VoIP security assessment
Netconf overview

- XML-based Network management configuration protocol
- Transport protocol
  - SSH, SSL, BEEP, SOAP, ...
- RPC
  - get, get-config
  - edit-config, copy-config, delete-config
  - lock, unlock
  - Close-session, kill-session
- Multiple configurations
  - Running
  - Candidate
  - Startup
- Capabilities
  - Writable-running
  - Xpath
  - Candidate
  - ...

18th NMRG – VoIP Security Assessment
Abdelnur, Cridlig, Bourdellon, State, Festor
YencaP architecture
YencaP modules system

Asteriks_Module
VoIP

Module
+ Public copy-config
+ Public get-config
+ Public edit-config

register to
Module Resolver

Firewall_Module
+ Public Method
  # Protected Method
  – Private Method

Interface_Module
+ Public Method
  # Protected Method
  – Private Method

Easy_Module
+ Public Method
  # Protected Method
  – Private Method

BGP_Module
+ Public Method
  # Protected Method
  – Private Method

RBAC_Module
+ Public Method
  # Protected Method
  – Private Method

XML configuration

Module Register Table

<table>
<thead>
<tr>
<th>Module</th>
<th>XPath expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firewall_Module</td>
<td>/security/firewall</td>
</tr>
<tr>
<td>BGP_Module</td>
<td>/config/network/bgp</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Manager web interface

- Agent list
- Operations
- Parameters
- Help
Access control on VoIP

<get-config>
/netconf/asterisk/data
</get-config>

<netconf>
<asterisk>
<data></data>
</asterisk>

<netconf>
<asterisk>
<data>
</data>
</asterisk>

Netconf Agent

m1, read, /netconf/asterisk/data

Access control policy
Asterisk overview

- Open source pbx system developed by Mark Spencer
- Supports VoIP with SIP, h323, and IAX
- Linux based
Mapping configuration files to XML

NetConf Module

<?xml version='1.0' encoding='utf-8'>
<asterisk>
  <file name="asterisk.conf">
    <section name="global">
      <attribute name="astetcdir">/etc/asterisk</attribute>
    </section>
  </file>
  <file name="sip.conf">
    <section name="general">
      <attribute name="context">netconf</attribute>
      <attribute name="realm">mydomain.tld</attribute>
      <attribute name="port">5060</attribute>
      <attribute name="bindaddr">0.0.0.0</attribute>
      <attribute name="srvlookup">yes</attribute>
    </section>
  </file>
</asterisk>
YencaP Module

NetConf Manager

Request

NetConf Agent

NetConf Module

NetConf Module

For Asterisk

NetConf Module

Response

Asterisk PBX
NetConf Demo

Add a new user

username="FOO"
password="BAR"
Fuzzy Packet is a tool that is able to manipulates and generates data messages by injecting and capturing packets into a network.

Its functionality depends on the XML templates which configure the actions to take.

Its architecture embeds on a Plug-In model, so it provides an easy extensibility to new features.
Data Generator (Fuzzer)

Random user and password generator
PlugIn

<table>
<thead>
<tr>
<th>Generates</th>
</tr>
</thead>
<tbody>
<tr>
<td>REGISTER sip:PBX-Server.com SIP/2.0</td>
</tr>
<tr>
<td>CRLF</td>
</tr>
<tr>
<td>From: sip:</td>
</tr>
<tr>
<td>variable user</td>
</tr>
<tr>
<td>string @PBX-Server.com</td>
</tr>
<tr>
<td>CRLF</td>
</tr>
<tr>
<td>Call-ID:</td>
</tr>
<tr>
<td>randomstr</td>
</tr>
<tr>
<td>minlen 5</td>
</tr>
<tr>
<td>letters/</td>
</tr>
<tr>
<td>numbers/</td>
</tr>
<tr>
<td>randomstr</td>
</tr>
<tr>
<td>string @PBX-Server.com</td>
</tr>
<tr>
<td>CRLF</td>
</tr>
</tbody>
</table>

Other possible actions
- Replace Reg. Expressions
- Repeats blocks
- Conditionals, probabilities
- Execution of Python Methods

REGISTER sip:PBX-Server.com SIP/2.0
From: sip:Smith345@PBX-Server.com
Call-ID: 9F1994A81DD@PBX-Server.com
User Enumeration

Captures the replies and generates an authenticating message if necessary

Generates REGISTER messages and inject them into a network.

Assign the captured packet to the msg variable

Check if the Call-ID is in our list of msg

Check if Status-Code is 100

Trying. Add to a priority list

Check if Status-Code is 200

Ok. User accepted by the Server

An Authentication challenge is required, so it inject a reply message for it

<filter>udp and port 5060</filter>
<capture>
  <assignvar>msg</assignvar>
  <continue>callidExist(msg)</continue>
  <choice>
    <option>
      <condition>
        isUserExists(msg)
      </condition>
    </option>
    <option>
      <condition>
        isUserAccepted(msg)
      </condition>
    </option>
    <option>
      <condition>
        isRequiredAuth(msg)
      </condition>
    </option>
  </choice>
  <injectreply>
    <invertpacket/>
    <assignvar>SIP/Auth.xml"/>
    </injectreply>
  </capture>
Spam Injection

Listening RPT packets

Injecting Spam
ARP Injection
(achieving “man in the middle”)

We want to see all the packets from a Computer A.

- Reply the ARP Request send by every computer involving the IP of A.
- To computer A: every IP go through the intruder
- To other computers: A's IP goes through the intruder

Still, we want to renew the cache before they send a Request message, so we can send ours, once in a while to ensure they send all packet through our the intruder
Demo

TestBed

Called party

Caller party

PBX

DHCP

TFTP

SWITCH

ARP Injector

Injests ARP packets into the caller in order to be in the middle of every packet it send.

Spam Injector

Injests RTP packets (a recorded message) into the called party conversation.

Listener

Capture RTP packets from the called party conversation and play them in the current computer.

User Enum

Try to register (brute force) in the PBX by injecting and capturing packets in the network.