An overview of some techniques to exploit VoIP over WLAN

Gianluigi Me, Damiano Verdone
University of Rome “Tor Vergata”
me@disp.uniroma2.it, damiano.verdone1@tin.it

Abstract

Voice over IP (VoIP) is one of the most emerging technologies, with a very relevant market penetration trend. This technology will represent a cost advantage for the business and private networks with greater flexibility, if no new related vulnerabilities are introduced. The problems of security of the VoIP are mainly related to the weaknesses of the combination of the SIP and RTP protocols. In the VoWiFi case, these weaknesses are enhanced by the intrinsic vulnerabilities of the first generation wireless networks (802.11b), or by a bad administration of wireless security systems. After building a VoIP network over WiFi without enforcing security measures for the authentication and the privacy of the data, we will show in this paper several typologies of attack: Eavesdropping and Sniffing of the VoIP calls, Man in the Middle, Denial of Service, Call interruption and Build false calls. All these threats can represent part of a check list for a plug-and-play penetration test schedule, whenever a company deploys a VoIP network infrastructure based on some untested VoIP softphone and Wireless Lan (as an internal hotspot).

1 Introduction

The aim of the present paper is to analyse the security problems when you realize VoWiFi systems, showing the practice of the intrinsic vulnerabilities of wireless networks and VoIP protocols and how they can be exploited to attack the privacy and the reliability of the VoWiFi communications.

The paper also proposes a solution to defend the system from exposed threats. Assuming however that a "valid recipe" able to give back an inviolable and safe network does not exist, but the security mechanisms to be used change according to the cases on the basis of the network architecture and the purposes the network is used for.

In the next sections we will describe various typologies of attack which exploit the vulnerabilities of a VoIP over WiFi network not adequately protected. Several typologies of attacks were tested on this network: Eavesdropping and Sniffing of the VoIP calls, Man in the Middle, Denial of Service, Call interruption and false calls building. Finally, we will present some countermeasures as mandatory security level, even if, in some cases, introducing security disabling some technology functionalities. For example, introducing firewalls to a VoIP network complicates several aspects of VoIP, most notably dynamic port trafficking and call-setup procedures.

2 Related work

Even if the technology and the VoIP protocols are not recent, the adoption of this technology has been knowing a wide diffusion only lately. For this reason there aren’t still many studies on the security problems of the VoIP and VoWiFi.

The NIST in the special publication 800-58 treats deals security consideration for Voice over IP System. Further vulnerabilities are exposed in [1], [2], [3].

The article underlines that if mobile units are to be integrated with the VOIP system, products implementing WiFi Protected Access (WPA), rather than 802.11 Wired Equivalent Privacy (WEP) have to be used.

3 Eavesdropping and Sniffing of the VoIP calls

The eavesdropping of the telephone calls, and their consequent dispatch to the parts involved in the communication is one of the most impressive attacks to the VoIP. It’s possible to mount an MITM attack in a wired network via well known techniques, e.g., using an ARP poisoning attack to force the SIP proxy, and the VoIP telephones to lace a communication with a malicious third party and not with one among them [4]. The high relevance of this attack technique has been stated, e.g., by the US Federal Communications Commission (FCC), which, in August 2005, ruled that broadband VoIP must comply with Communications Assistance for Law Enforcement Act (CALEA) [5].
In the wireless environment, given the nature of the transmission means, in the security conditions described before, the interception of the VoIP calls is even simpler. Anyone supplied with a laptop, with an appropriate wireless adapter and a sniffer software, can listen to a VoIP call which takes place on an unprotected WiFi network. The Ethereal sniffer supports specific functions to identify the VoIP calls in the sniffed packets, using the SIP protocol, and to rebuild the audio stream starting from the payload of the captured RTP packets. Ethereal is able to rebuild graphically all the exchanges between the two communication parties in the selected call (Figure 1).

Furthermore, Ethereal is able to identify the various RTP stream in the captured packages, enabling the extraction of the vocal content of the various packages, rebuilding the whole conversation and saving it in an audio file ".raw" or ".au".

4 Man in the Middle

On the switched wired networks, man-in-the-middle attacks are frequently used to allow the possibility of traffic sniffing. 802.11 LANs are shared medium networks by definition, and once you’ve dealt with the encryption (e.g. WEP) you can sniff all the packets on the LAN even without being connected to it. Once “in the middle” of two wireless hosts, it’s possible to inject commands into the traffic streams between both hosts. Man-in-the-middle attacks on WLANs can occur on both the first and second OSI layers. Layer 1 man-in-the-middle attacks refer to jamming an existing wireless AP while providing your own clear signal AP at least five channels away from the attacked AP channel. The jamming can be performed using a specific jamming device or by flooding the AP channel with junk traffic.

Of course, the parameters of your rogue AP should reflect the parameters of the legitimate access point. Layer 2 attacks differ by using a spoofed deassociation or deauthentication frames flood to kick the target host from its link with a legitimate AP. This is generally more efficient than the channel jamming. A determined attacker can easily combine both Layer 1 and Layer 2 attacks to reach the maximum effect.

The majority of modern client cards will detect the new rogue AP on a channel different from the one they currently use and automatically associate with it, if the association with the legitimate AP becomes hard or impossible.[6]

4.1 The Man-in-the-Middle attack in practice

The aim of the attack is to interpose between a determined wireless client and the legitimate AP, in a transparent way both for the wireless client and the legitimate AP.

Two wireless adapters installed on the same computer are necessary to execute this attack. We used the adapter IPW2200b/g, and the wireless adapter Dwl-G650, whose suitable driver exists for IPW2200b/g (http://sourceforge.net/projects/ipw2200-ap), enabling IPW2200b/g to work like an access point. We built up the rogue access point to deceive the wireless client exploiting this driver. For the wireless adapter Dwl-G650, the new driver Madwifi allows the wireless card to operate in master mode, also creating several wireless virtual interfaces, which can simultaneously work in various modes.[7]
legitimate access point on its channel is very little, the rogue access point works on a different channel to raise the attack performance. The station establishes on the basis of the signal power perceived, to which between the two access points to associate it. You can use the mechanisms described in the beginning of this section, attack at level 1 (Jamming) and attack at level 2, to degrade or even cancel the coming signal by the legitimate access point.

The procedure to be followed for the attack to be executed is the following:

```
#!/bin/sh
#
wlanconfig ath0 destroy
wlanconfig ath0 create wlandev wifi0 wlanmode monitor
wlanconfig ath create wlandev wifi0 wlanmode sta
nosbeacon
iwconfig ath1 essid default
iwpriv ath1 wds 1
iwconfig eth2 channel 9 essid default
brcctl addbr br0
brcctl addif br0 eth2
brcctl addif br0 ath1
ifconfig eth2 0.0.0.0 up
iwconfig ath0 channel 1
ifconfig ath0 up
airplay -0 19003 -a 00:0f:3d:0b:51:b4 -c 00:11:d8:b3:32:f7
ath0
```

The card DWL-G650 and the driver Madwifi create two network logic interfaces doing one in monitor mode (ath0) and the other in managed mode (ath1). Both work at the channel 1, where also the legitimate access point works. The adapter IPW2200b/g (eth2) works on the channel no. 9 with ESSID "default" as the legitimate AP.

Furthermore from the previous listing, the presence of a wireless device in addition to the DWL-G650, because the logical interfaces created with the driver Madwifi cannot act on different channels, unless they operate in monitor mode.

The sequence of the commands creates, a WDS (Wireless Distribution System), between "ath1" and the legitimate AP. WDS support allows "ath1" to manage 802.11 four-address format frame.

Furthermore from the previous listing, the presence of the command "brcctl" allows the creation of an Ethernet bridge, which allows to join different Ethernet network segments [8]. The last lines of the script contain the configuration of the interface "ath0" and the execution of the tool aireplay which does part of the aircrack suite. Airplay produces and injects in the wireless network through "ath0" the de-authentication frame, which should disconnect the client from his legitimate AP and degrade the quality of the signal on the channel no. 1.

There may be many reasons why a station links to a rogue access point (set up with the same ESSID of the legitimate AP) exchanging it for the legitimate AP. As both the access points work on the same channel, the main cause is the greater access point signal power compared to that of the other. Unfortunately you will hardly be in the right place, or have the adequate equipment to dominate the signal of the legitimate AP.

Furthermore, disabling the dispatch of beacon frame to hide the ESSID of their network is completely useless, since many other exchanged frames contain the ESSID.

If the network manager deactivates beacon frame dispatch on the legitimate AP, the performance of the previously described attack will increase enormously, even if the signal power of the legitimate AP is higher than that of the rogue access point. Fewer de-authentication frames are needed. However, if the two AP are on the same channel remains, very likely the result of the attack will be a Dos caused by the RF interference of the rogue AP.

The wireless client who can’t access the network from the channel no. 1, will check also the other transmission channels. If you are lucky it will find at channel no. 9 an access point reflecting the pre-arranged parameters to associate itself.

If the legitimate AP sends beacon frames, during the period where the station is associated with the rogue access point, the de-authentication frames must keep on being injected on the active communication channel.

Setting the bridge between the wireless station and the access point let the attacker to easily observe all the packets which leave from and arrive to the client and inject any type of packet inside the network. Now the eavesdropper has the complete control over the VoIP traffic between the stations. For example, he can decide to filter the exchanged packets. In Linux environment this operation can be performed using Netfilter/iptables.

For instance, observing the VoIP packets in transit, you can easily determine the UDP port through which the call is set up through the iptables command, as the following:

```
nftables -A FORWARD -s 10.0.0.4 -p udp --source-port 5070 -j DROP
```

5  DoS - Denial of Service

DoS attacks further threaten the VoIP service. A SIP service (or a client, or a SIP server) can fail, because of an SIP not valid messages, with well-known vulnerabilities [9]. If a server does not have mechanisms to manage (or simply ignore) the incorrect messages, it could offer vulnerabilities to exploit.

PROTOS Suite Test can verify the client and SIP server behaviour, checking what every PBX (Private Branch Exchange) owner should execute. PROTOS uses a subset
of the SIP messages, mainly INVITE messages, to find out and evaluate the vulnerabilities of the servers, and above all of an SIP client [10].

The test consists in dispatching ad-hoc suitably created SIP messages (Test-Case), containing syntactic mistakes. A syntactic mistake is an exceptional element, data which are suitably planned to cause an unwelcome behaviour. A single case of test can contain one or more exceptional elements, which can violate the specifications of the protocol.

After capturing the SIP setup messages of a SIP call and finding out the used client SIP, generally shown in the field of the header "User-Agent", the softphone can be identified using PROTOS, to find vulnerabilities to be exploited. The tools checked for vulnerabilities are CallConductor v. 1.03, Express Talk v.1.03 and X-lite 1103. The clearest problems are:

- Express Talk v.1.03: the INVITE messages with the Content-Length negative value cause the crash of the program;
- CallConductor v. 1.03: as Express Talk;
- X-Lite 1103: an INVITE message with a Content-Length value equal or higher than 1073741823 bytes causes a remarkable increase in the use of the resources in the system above all the virtual memory.

If the vulnerability scanning task is successful for a certain softphone or PBX, you can attack the station or the wireless stations which use that softphone exploiting the found vulnerabilities. You can execute the attack avoiding the direct connection to the network and using the script python Wifitap.py based on the Scapy library. Wifitap allows direct communication with an associated station to a given access point directly, meaning:

- not being associated ourselves;
- not being handled by an access point

The communication over WiFi networks uses traffic injection [11].

E.g., in a wireless network using the softphone Express Talk, you can send every terminal a message SIP INVITE with Content-Length negative value. The Express Talk application will crash. Using X-lite, you can send an INVITE SIP message with a Content-Length value equal or higher than 1073741823 bytes to each network host causing a dramatic decrease of performance.

Furthermore, through Wifitap it’s possible to mount storm-invite or storm-register attacks against softphone and PBX: for instance, we realized an attack of this type against Express Talk and X-lite using SIPp, a free Open Source test tool/traffic generator for the SIP protocol. A storm-invite attack against Express Talk causes a CPU usage of 100%, when the call rate is equal to about 1347 calls every two seconds.

### Exceptional Element Categories

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>empty</td>
<td>Omitted (empty) element content</td>
</tr>
<tr>
<td>ipv4-ascii</td>
<td>Exceptional IPv4 addresses in ascii</td>
</tr>
<tr>
<td>overflow-general</td>
<td>'a' (0x61) character overflows up to 128k</td>
</tr>
<tr>
<td>overflow-slash</td>
<td>Overflows of '/' up to 128 kbytes</td>
</tr>
<tr>
<td>overflow-colon</td>
<td>Overflows of ':' up to 128 kbytes</td>
</tr>
<tr>
<td>overflow-space</td>
<td>Overflows of ' ' up to 128 kbytes</td>
</tr>
<tr>
<td>overflow-null</td>
<td>Overflows of 0x61 and nulls (0x00) mixed</td>
</tr>
<tr>
<td>overflow-leftbracket</td>
<td>Overflows of '&lt;' up to 128k</td>
</tr>
<tr>
<td>overflow-rightbracket</td>
<td>Overflows of '&gt;' up to 128k</td>
</tr>
<tr>
<td>overflow-at</td>
<td>Overflows of '@' up to 128k</td>
</tr>
<tr>
<td>overflow-equal</td>
<td>Overflows of '=' up to 128k</td>
</tr>
<tr>
<td>fmtstring</td>
<td>Format strings (e.g. %s%s%s or %.4097d)</td>
</tr>
<tr>
<td>utf-8</td>
<td>Malformed UTF-8 sequences</td>
</tr>
<tr>
<td>integer-ascii</td>
<td>Pos/Neg ASCII encoded integers</td>
</tr>
<tr>
<td>ansi-escape</td>
<td>Malformed ANSI escape sequences</td>
</tr>
<tr>
<td>sip-version</td>
<td>Malformed &quot;SIP/2.0&quot;</td>
</tr>
<tr>
<td>content-type</td>
<td>Malformed &quot;application/sdp&quot;</td>
</tr>
<tr>
<td>sip-URI</td>
<td>Malformed SIP-URI</td>
</tr>
<tr>
<td>sip-tag</td>
<td>Malformed tags</td>
</tr>
<tr>
<td>crlf</td>
<td>Arrangements of CR (0x0d) and LF (0x0a)</td>
</tr>
</tbody>
</table>

Table 1: PROTOS, Exceptional Element Categories [10]

### 6 Call interruption

Furthermore, the forwarding of a BYE message would take the participant of a call to the immediate interruption of the call itself. First of all the aggressor must know the Call-ID of the Call-Dialog. The Call-ID acts like an univocal identifier to assemble a set of messages. This must be true for all the requests and the response sent by both the UAs in a dialogue. Therefore, the aggressor must intercept the call setup to be able to interrupt it using the appropriate value for the Call-ID field [4].
Another method to interrupt a call is to use the CANCEL method. We build a Python script to implement this attack, which opens a socket, using the scapy library [12], on a wireless interface in monitor mode. The script analyses all the captured traffic to detect SIP setup, in particular the INVITE messages.

When the attacker (while eavesdropping) detects a SIP INVITE message in the traffic, forges an ad-hoc SIP CANCEL, which contains the same Call-ID as well as other tag in the INVITE request. The so built CANCEL message, will be injected in the wireless channel, as shown in the figure 3. The destination and source address of the injected packets, both at layer IP and layer MAC are the same of the INVITE request, and also the BSSID is obviously the same.

Building false calls

An attacker can send two appropriate forged SIP INVITE messages, respectively to two target User Agents, establishing a call between them, without any action performed by the unaware victims. The two forged INVITE messages have the same values for the fields Call-ID, CSeq and for the Branch parameter (transaction identifier), but in the two messages, source and destination will be swapped. The body of the INVITE request will have to contain according to the SDP protocol, the same information about the audio codec which you want to use during the call. Both the softphones must be able to produce an audio stream using that codec. A simple analysis of the RTP traffic can reveal the starting value of the UDP port (8000 commonly used) and as the softphones increase it for next calls.

After sending the two INVITE messages, the attacked devices start to ring and they send SIP 180 Ringing messages to the one supposed to be the sender of the INVITE request. The receiving station cannot understand, so it will discard them. When the user answers the call, the softphone sends a SIP 200 OK response message always to the one who it considers to be the responsible for the INVITE request, also this message is however discarded by the receiver.

To successfully complete this operation, when the attacker detects a 200 OK message in the captured traffic, which refers to the false call (this is established on the basis of the value of the Call-ID), injects an acknowledgment packet containing the same source and destination fields of the previous INVITE request, as well as the same value for the Call-ID field.

As for the previous cases, we built a python script using the scapy library. Most SIP softphones are able to manage at least three calls simultaneously, therefore the script tries to establish at least three new calls between a terminal and other three which are present in the same WLAN. Even if seldom, it can happen that the setup process of a false call is successful, but one or both the terminals of the communication do not receive the audio. That is due to a discrepancy between the UDP ports actually used for the RTP streams and the ones agreed during the false setup. The RTP packages will arrive to destination, but not on the port where the softphone is in listening and they will be discarded. However both the terminals will produce a RTP
An attack of this type produces many contemporary SIP calls inside the network. Even if of short duration (the time to understand that none of the two parts began the call), these calls cause a considerable increase in the traffic inside the network. Furthermore all the listing terminals in the configuration file will not be able to receive calls because occupied to manage some conversations which are not wished.

8 Solution

A security mechanism able to guarantee the privacy, and the authentication of the data is necessary. The chosen security mechanism is the WPA standard with the protocol of 802.1 X/EAP authentication. The cryptography of the WPA standard protects the system from eavesdropping of the VoIP calls, and intercepting calls setup (Call-ID, and header User-Agent), while the protocol of 802.1X/EAP authentication protects the VoWiFi network from the packets injection in the network and from Man in the Middle attacks. An authentication server (radius server) was added to the network architecture (http://www.freeradius.org). The chosen authentication mechanism was "EAP-TLS", which use X.509 digital certificates, EAP-TLS allows the mutual authentication between the client and the AP [13].

9 Conclusion

In this paper we show that the privacy of the communications VoIP on WLAN represents a major security concern: anyone supplied with a laptop, with a wireless adapter and software, freely downloadable on the Web, is able to listen to a VoIP call which takes place on a WiFi not (or badly) protected network.

Even the effectiveness of the whole network is in danger. Since the coexistence of voice and data on the same network is one of the biggest company advantages of the adoption of the VoIP technology, we show how to mount attacks exploiting the vulnerabilities of the signaling protocol SIP in connection with a wireless environment that aim at compromising the whole operation of the network: not only the vocal communications, but also the exchange of data, saturating the band of the systems. The majority of the attacks will exploit software vulnerabilities (e.g. if the proxy SIP or the UAC (User Agent Client) do not elaborate the Call-id correctly or if the attacker is able to intercept the Call-ID). Unfortunately the SIP messages are codified with the UTF-8 standard and can be read easily.

Since VoIP adoption is related to a cost cutting plan, many companies, reasonably, can adopt free softphones as company VoIP infrastructure clients, exposing the VoIP service at least (and the company network) to the attacks we shown in this paper.

Furthermore, extending the VoIP network to wireless 802.11b segments represents a major concern for security and privacy of communication. In conclusion, it’s mandatory to implement security mechanisms (e.g. 802.11i/AES with EAP) to guarantee security and privacy and mutual authentication for VoIP company services.

References

[1]. Vulnerability Note VU#749342, Multiple vulnerabilities in H.323 implementations, www.kb.cert.org/vuls/id/749342
[8]. Uwe Böhme. Linux BRIDGE-stp-HOWTO. Revision v0.04, 11 January 2001, Revised by: U.B.