SIPERA VIPER LAB

1900 Firman Drive,
Suite 600
Richardson, TX 75081
USA

T 214-206-3210
F 214-206-3215

Attacking VoIP through IPSec Tunnels

Sachin Joglekar

Sipera VIPER Lab White Paper

Black Hat DC Briefings 2008
February 20-21, 2008
Washington DC, USA

1900 Firman Drive,
Suite 600
Richardson, TX 75081
USA

T 214-206-3210
F 214-206-3215

Table of Contents

1900 Firman Drive,
Suite 600
Richardson, TX 75081
USA
T 214-206-3210
F 214-206-3215

Introduction

With 1 in 8 homes in the US using a cell phone as their primary and only phones[1], a cell phone integrated with VoIP over WiFi service (a dual-mode phone), which allows them to use their cell phones from any wireless access point without using cellular minutes, would be a major benefit to home and business users! This VoIP service over home/office wireless Internet also has better coverage inside buildings, eliminates the need to have two separate numbers, and can provide free international roaming.

In the context of mobile networks, such dual-mode phones switch calls between the cellular network and the IP network depending on the signal strength. When switching a call to the IP network, or just simply initiating a call on an IP network, the core IP telephony servers need to authenticate the phone over the IP network before it can allow the traffic from the phone to be routed elsewhere.

EAP-SIM [3] is one such authentication mechanism used in the wireless networks and widely adapted to the greater security needs of VoIP networks. Additionally, an encrypted communication channel (such as an IPSec [4] Tunnel) is used between the mobile phone and the core IP telephony network. These security mechanisms are implemented to allow traffic from only the authenticated endpoints. It is assumed that the authenticated endpoints are trusted and will not misbehave. However, as we will show in this paper, such trust can be exploited to launch attacks on the core IP and cellular telephony networks. Such attacks could manifest as SPAM, DoS, and theft of service on billions of non dual-mode cell phone users which has devastating impact on communication infrastructure.

Current security framework in mobile VoIP networks

To seamlessly communicate between traditional cellular phones and phones connected over the IP network, core networks have been adapted to transcode the signals between the two networks. Typically, among other gateways, this involves adding a security gateway and an IP telephony server to the core network while leveraging the existing authentication infrastructure. The security gateway facing the Internet is responsible for enforcing the authentication of the phones and encrypting the VoIP signaling and media traffic, before forwarding the traffic to the core servers and gateways.

In GSM cellular network that is most widely deployed in the world, such authentication is done using the Subscriber Identity Module (SIM) card associated with the phone. However, the security strength offered by the GSM authentication is insufficient when authenticating the dual-mode phone over the IP network. As a result, EAP-SIM (Extendible Authentication Protocol-SIM) [RFC 4186] authentication mechanism was standardized by IETF and is widely used for authentication and session key distribution for dual-mode GSM/VoIP phones. EAP-SIM provides keys of greater strength by using

the 64-bit key to derive additional keying material. Additionally, EAP-SIM supports other features like user anonymity, result indications, and fast re-authentication. This increased security strength is used to secure VoIP communication between the dual-mode phone and the core IP telephony network by establishing what is called an IPSec tunnel between them. IPSec is a suite of protocols [4] for securing IP communications by authenticating and/or encrypting each IP packet in a data stream [5]. IKEv2 (Internet Key Exchange v2) protocol, which is a component of the IPSec protocol suite, uses EAP and EAP-SIM to perform mutual authentication and establish and maintain security associations (SAs).

Figure 1 shows an example of a dual-mode phone connecting to a security gateway using an IPSec tunnel. As seen there, the security gateway is expected to protect the internal telephony core network by allowing only traffic coming through the IPSec tunnels from authenticated legitimate phones.

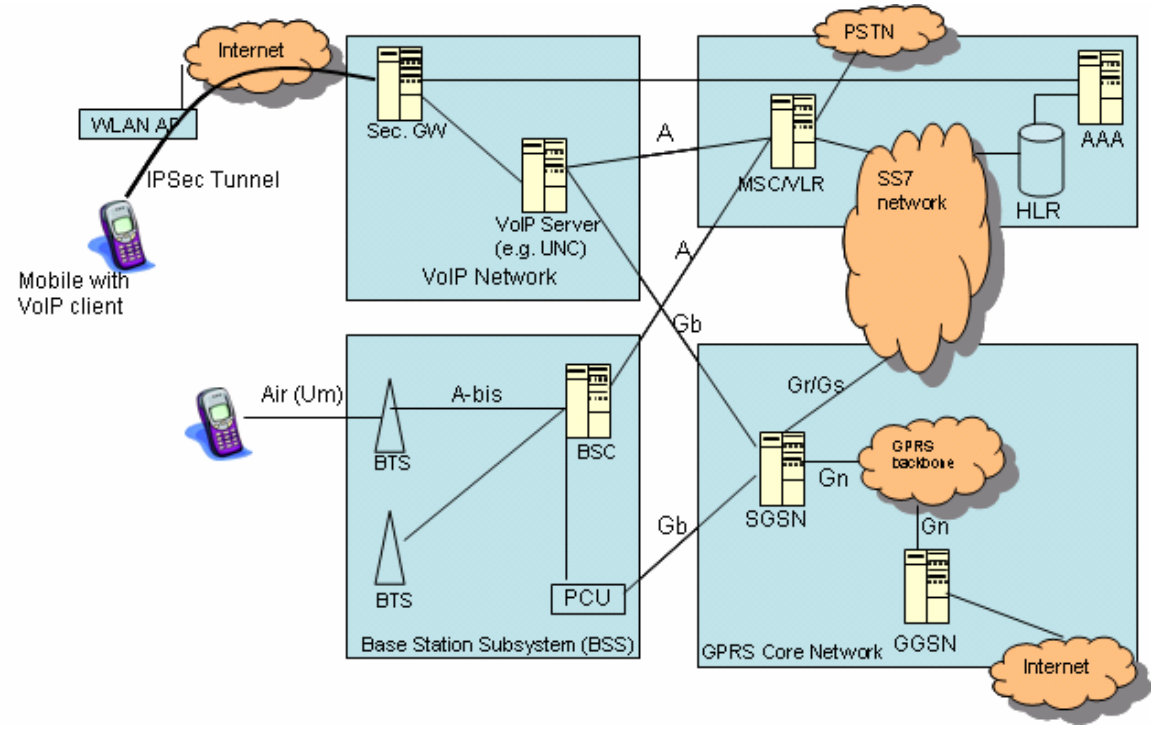

Figure 1: Reference network architecture

Figure 2 shows this call flow in more detail by outlining the IPSec messages exchanged between the dual-mode phone and the security gateway and the interactions with the SIM module in the phone. The SIM card is in fact inside the dual-mode phone, but for the purpose of clarification, we have shown it separate. As seen in Figure 2, the VoIP core challenges the phone in step 6 with a 128-bit random value (typically more than one challenge is sent). The phone firmware then invokes the GSM algorithms to calculate the response to the challenge (steps 7 and 8) and eventually an IPSec tunnel is setup between the phone and the VoIP core and all subsequent traffic, to and from the VoIP core, passes

through this tunnel. It is assumed that, since only after successful SIM based authentication the mobile phone is allowed to use voice services, any traffic coming through the IPSec tunnel can be trusted. However, as discussed in the subsequent sections, this assumption proves dangerous to the security of the entire VoIP and cellular core network and its subscribers.

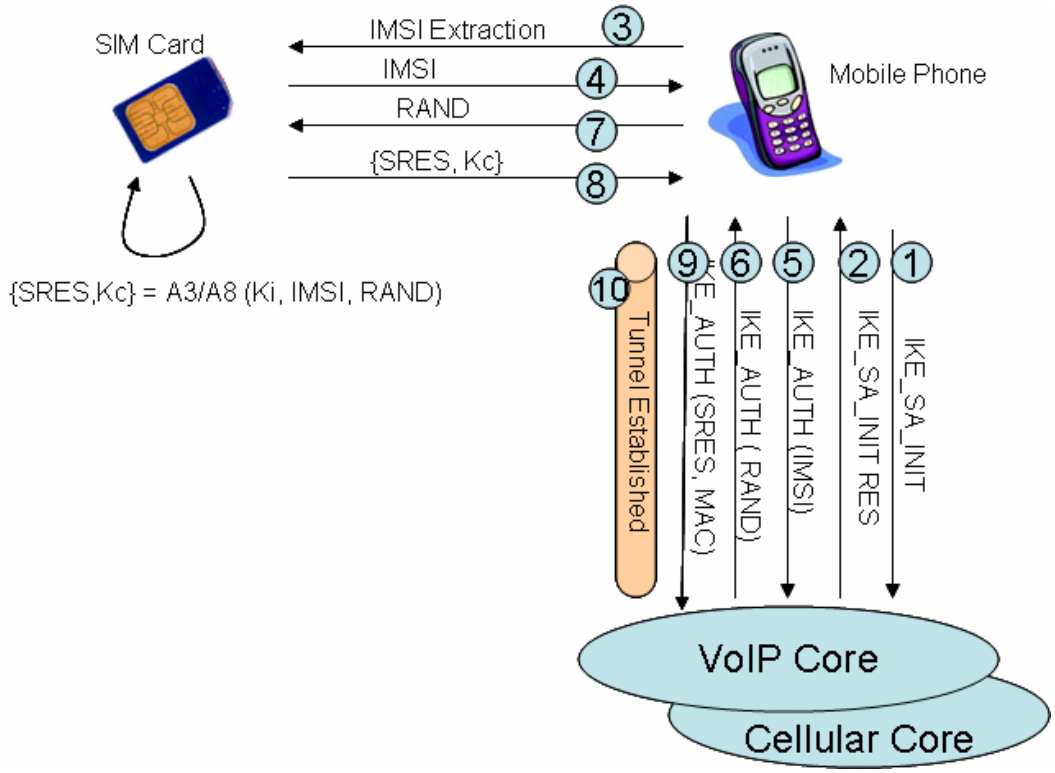

Figure 2: Call flow of setting up IPSec tunnel using EAP-SIM

Can your laptop become a legitimate phone?

From the perspective of the security gateway deployed in the VoIP core network, what is a legitimate dual-mode phone? Is it any endpoint that can respond to the challenge successfully and setup an IPSec tunnel? The answer is clearly yes as the security gateway relies on the SIM card for the authentication not the device itself. But, what if we use a laptop, respond to the challenges and setup an IPSec tunnel? The VoIP core will have no way to distinguish between traffic that is coming from a legitimate phone vs. traffic coming from a laptop that is able to use SIM credentials.[ξ]

[ξ] This flexibility allows cellular providers to offer soft phones which is a big value in integrating with desktop applications.

In this paper we show how a hacker can use following tools and gadgets to establish an IPSec tunnel to the security gateway and launch attacks.

1. One Linux laptop
2. A GSM phone with IP telephony feature
3. A valid SIM card
4. A USB SIM card reader
5. SIM card reader interface software
6. IKE daemon to establish IPSec tunnel

What happens inside the phone?

In order to get an IP address associated with the security gateway, the phone first needs to authenticate itself to the security gateway. Figure 3 shows full EAP-SIM authentication procedure as given in RFC 4186, where additional details of this call flow are available.

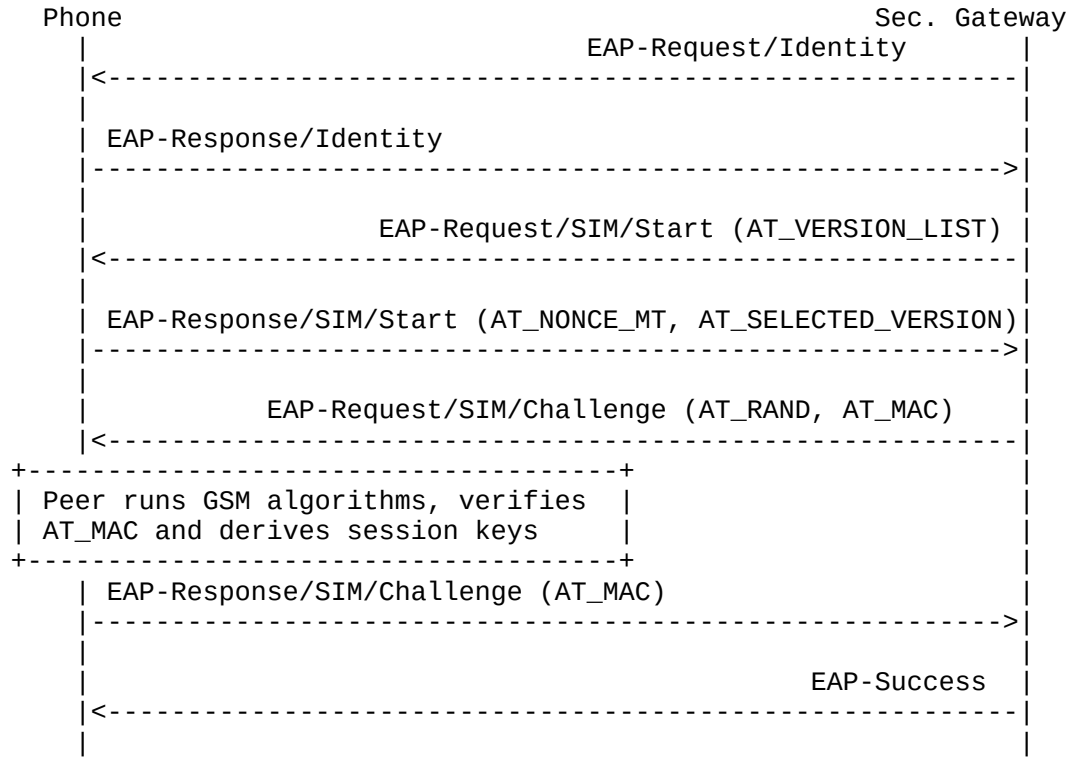

Figure 3: EAP-SIM full authentication procedure (RFC 4186)

To authenticate the phone, the security gateway sends one or more challenges (AT_RAND) to the phone. These challenges are typically 128-bit random numbers generated as a part of GSM triplets received by the security gateway from a backend EAP server. This 128-bit challenge is a component of what is called a GSM authentication triplet. Other two components of a GSM triplet are a 32-bit response (SRES) and a 64-bit

key (Kc). The phone provides the RAND challenge to the SIM card through the SIM card interface and invokes the GSM authentication algorithms on the SIM card. Based on the secret that is stored in the SIM card, the SIM card then generates the other two components of the triplets: 32-bit response (SRES) and 64-bit key (Kc). SRESs is hashed with some other fields and sent to the security gateway as a response to challenges. In the GSM 2G mobile network standard, Kc was originally intended to be used as an encryption key over the air interface. But, as discussed before, 64-bit key strength is not sufficient for IP networks and hence the Kc is used to derive the keying material. Section 7 of RFC 4186 describes how the keying material is derived.

For our purpose, it is important to understand that responding to the challenge is made possible by the phone firmware that is able to interface with the SIM card and access the keying material generated. If we are to take the SIM card out of the phone and connect it to a USB SIM card reader, all we need is a program that can do the job of the phone's firmware, i.e. successfully interface with the SIM card, extract the triplets and IMSI from the SIM card, and successfully respond to the challenges from the security gateway to establish an IPSec tunnel. The following section will show how we can achieve this.

Putting the tools and gadgets together

SIM card reader devices fall in a general category of smart card readers and there are several of them available. To use the SIM card reader, we need software which interfaces with the SIM card through the reader. As mentioned earlier there are free tools available which can be enhanced and used with SIM card readers to access the authentication triplets from the SIM card. One such tool is PC/SC-Lite [Project page: https://alioth.debian.org/projects/pcsclite/, Documentation: http://pcsclite.alioth.debian.org/pcsc-lite/] library which is middleware to access a smart card using SCard API (PC/SC) from a Windows or a Linux box. This library provides API functions to establish communication context with the smart card, connect to/disconnect from the card, and query the status and other attributes of the card among others. It supports both T=0 and T=1 protocols (More info: http://www.sat.su/satxpress/SmartCard/ISO7816-3.htm). However, the PC/SC-Lite library does not provide an API to read the authentication triplets generated by the SIM card as part of the EAP-SIM. For this demo, we have extended the library to add this functionality (See Appendix A for these enhancements).

The next requirement is a tool to negotiate and establish IPSec tunnel to the security gateway. Again, there are several tools available for this purpose and we chose racoon (http://ipsec-tools.sourceforge.net/) for this demo. Racoon implements the IKE key management protocol to establish security association with other hosts. However, to be able to use it with the SIM card we added EAP and EAP-SIM support to Racoon. These enhancements are also given in Appendix A. EAP is a universal authentication framework defined in RFC 3748 and can be used by two parties to negotiate authentication mechanism (e.g., EAP-MD5, EAP-TLS, EAP-IKEv2, EAP-SIM(2G), or EAP-AKA(3G)). Figure 4 shows how these components fall in place.

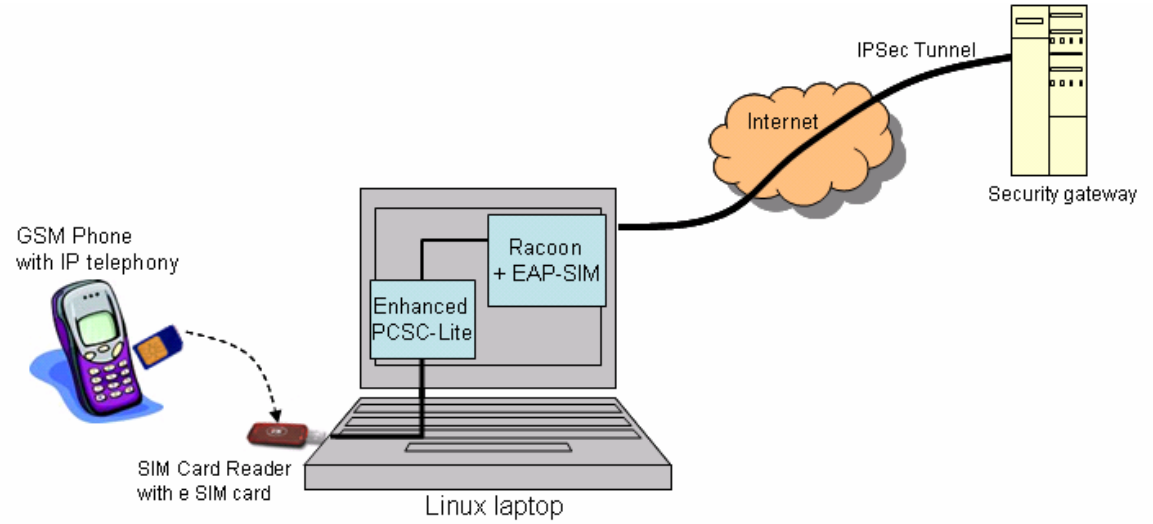

Figure 4: Putting all gadgets and tools together

By adding EAP-SIM support in racoon and adding triplet reading support in PC/SC-Lite, we can use a Linux box and a SIM reader to setup an IPSec tunnel with the security gateway. From the security gateway's perspective, it looks like a legitimate phone is establishing this tunnel since we are able to respond correctly to the SIM challenges using credentials stored in our SIM card. Now you can write a VoIP application, pump the traffic through the tunnel and launch a variety of attacks on the network and subscribers The reason: once the IPSec tunnel is successfully established the security gateway trusts all the traffic that is coming through that tunnel. All the security gateway does from a security perspective is terminate the IPSec tunnel but it blindly forwards all the internal traffic to the core network, whether it's legitimate traffic or attack traffic. The next section will discuss more about what attacks can be launched, on the core network, on other legitimate subscribers, and on the security gateway.

Attack!

As shown above, being able to successfully setup an IPSec tunnel with the security gateway using a laptop instead of a phone opens up several possibilities to launch attacks through that tunnel. Unlike the phone application that uses the tunnel to send voice traffic, the application on the laptop can be controlled by us and used to send any IP traffic. Now, there is nothing stopping us from writing a program to generate attack traffic and send it through the IPSec tunnel. We will take an example of a UMA (Unlicensed Mobile Access) that is commercially deployed and leverages dual-mode phone technology and discuss few attacks possible below:

Attacks on the core telephony servers

Referring to the reference network diagram shown in Figure 1, it is evident that in addition to a VoIP server, the core network contains cellular phone network servers which are as easily reachable through the IPSec tunnel as the VoIP server. Following are few sample attacks on the core telephony network that can be launched from our laptop posing as a legitimate phone.

IMSI Reconnaissance

The objective of this attack is to gather all valid IMSIs registered on the UMA server. This can be achieved by sending, through the IPSec tunnel, several Location Update requests with different IMSIs each time. The UMA server will either respond with Location accept or Location reject. Based on the response we can conclude if the IMSI is valid or not.

Media flooding

Objective of this attack is to exhaust ports on the media gateway. The media gateway is used to anchor VoIP media packets and convert the packetized voice into other types (PSTN, cellular, etc). On the IP side of the media gateway, there are limited numbers of ports available to anchor calls from subscribers who are talking on their phones over IP. These ports can be exhausted by flooding RTP packets through the tunnel towards the Media Gateway. Such port exhaustion will result in unavailability of voice channels to legitimate subscribers causing denial of service.

Network-wide Location Update spoofing

Objective of this attack is to corrupt the HLR database and cause denial of service to legitimate subscribers. Valid IMSIs discovered during IMSI Reconnaissance attack can be used to send Location Update requests with spoofed IMSIs. Since the security gateway forwards these requests to the HLR, the location records in the HLR database are overwritten with whatever location we supply.

Attacks on other legitimate subscribers

Denial of service using Spoofed Location Update

Objective of this attack is to corrupt the location record in the HLR server by spoofing the IMSI in the Location Update request. This will reroute the calls meant for the legitimate subscriber to us. Impact of such attack is denial of service to those legitimate subscribers whose IMSIs were spoofed.

Spam

This could be the most annoying attack on the legitimate subscribers and subscribers of other networks. After successfully establishing a call between our attack toolkit and another legitimate subscriber, we can play an automated unsolicited Spam voice message. Again, this attack is possible because, the security gateway and all other servers trust the traffic coming from us.

Attacks on security gateway

IKE_SA_INIT Flooding

Objective of this attack is to flood the security gateway with IPSec tunnel establishment requests. Even before the IPSec tunnel is established, the security gateway can be flooded with IKE_SA_INIT requests. Based on the maximum rate at which the security gateway can handle the tunnel establishment requests, we can either increase the CPU usage on the security gateway or completely exhaust this capacity.

IKE_SA_AUTH Flooding

This attack can penetrate up to, and including, the AAA server. Since for each IKE_SA_AUTH request, the security gateway requests the AAA server for credentials, flooding with IKE_SA_AUTH messages may overwhelm the AAA server.

Other attacks

In addition to the above sample attacks, there are several application level fuzzing and flooding attacks possible. Fuzzing attacks construct malformed protocol messages with an expectation of the protocol parser in the target system crashes disabling the server from processing further legitimate requests and causing denial of service.

Although it may be difficult to cause resource exhaustion on the high-end core servers by flooding from a single laptop, using multiple SIM readers and USB hubs it may be possible to increase the resource utilization of an already “working on the edge” server to tip it off. We have left this attack for future research.

1900 Firman Drive,
Suite 600
Richardson, TX 75081
USA
T 214-206-3210
F 214-206-3215

Conclusion

One of the primary advantages of Voice over IP (VoIP) is that it allows mobile operators and enterprises to extend their core telephony networks. And, with WiFi-enabled VoIP phones, users can connect to their core telephony servers over the Internet from any remote location. Often, such remote VoIP is secured using IPSec VPNs and, more importantly, many believe that all VPN tunnel traffic should be considered “trusted”.

In fact, it is quite easy to become an authenticated subscriber on the network and launch attacks on the core infrastructure. In this paper, we have shown how to exploit a SIM card from an IPSec VPN-enabled GSM/VoIP phone to launch attacks through the IPSec tunnel.

This demonstrates that IPSec VPNs are not sufficient to secure VoIP, and that it is possible to embed exploits inside the tunneled traffic to generate attacks on the core telephony network. In most cases, these attacks go undetected and can have a devastating impact on the core network. This needs to be a major concern to anyone rolling out VoIP and relying on IPSec only to secure the connection.

1900 Firman Drive,
Suite 600
Richardson, TX 75081
USA
T 214-206-3210
F 214-206-3215

Appendix A

Tools Released

SIMtool- SIM Interface to PCSC-Lite

PCSC-Lite provides a low level interface to access the smart card placed in the smart card reader. However, it does not directly provide functions to support the EAM-SIM authentication. SIMtool is a wrapper around the PCSC-Lite which does provide such functions. These include-

- Function to supply the RAND challenge from the EAP-SIM Authenticator server and get back the SRES and Kc.
- Function to unlock the SIM by providing the PIN number
- Function to read the IMSI from the SIM card
- Function to select the card reader reader (in case there are multiple readers attached)

Bottom line is, SIMtool allows you to pop out the SIM card from your GSM phone, connect the card to a laptop using a card reader, and run GSM algorithms on the card. Of course, this by itself is not sufficient to successfully establish the IPSec tunnel. For that, we would need an IKE tool with EAP-SIM support (which we talk about next).

Enhancements to Racoon tool to add EAP and EAP-SIM support

As we discussed earlier, to be able to make a laptop look like a legitimate phone to the VoIP core network, we need to support EAP-SIM authentication from an IKE/v2 client. Racoon (http://ipsec-tools.sourceforge.net/) is an IKEv2 daemon for automatically keying IPSec connections. However, it does not have support for EAP and specifically EAP-SIM. We are releasing some enhancements to Racoon which will add EAP and EAP-SIM support to it. Of course, these enhancements work with the SIMtool mentioned above.

Credits

Thanks to Dustin Trammell for his contributions to the simtool.
Thanks to Sudeep Patwardhan for his contributions to the Racoon enhancements.

References

1. New article: 1 in 8 Households Now Without a Wired Phone Line, *Dec 12, 2007*, http://tech.yahoo.com/blogs/null/61314?comment_start=6&comment_count=20

2. RFC 3748, *Extensible Authentication Protocol (EAP)*, IETF, http://www.ietf.org/rfc/rfc3748.txt

3. RFC 4186, *Extensible Authentication Protocol Method for Global System for Mobile Communications (GSM) Subscriber Identity Modules (EAP-SIM), IETF*, *http://www.ietf.org/rfc/rfc4186.txt*

4. RFCs 4301-4309, IPSec Protocol Suite, IETF

5. IPSec article on Wikipedia, http://en.wikipedia.org/wiki/IPsec

6. GSM Authentication, http://www.gsm-security.net/faq/gsm-authentication-key-generation.shtml

7. PCSC-Lite, *Middleware to access a smart card using SCard API (PC/SC)*, http://pcsclite.alioth.debian.org/

8. Racoon, *Internet Key Exchange (IKE) daemon for automatically keying IPsec connections*, http://ipsec-tools.sourceforge.net/