Exploits and the Attackers' Process

- Systems are attacked and compromised for a variety of reasons and to accomplish a number of objectives.

- It is important to understand the reasons and the purposes behind the attacks in order to deploy effective countermeasures against these attacks.

- There are several definitions of the word "exploit" depending on the context and the viewpoint. For example, one definition (www.dictionary.com) defines an exploit as "a notable achievement". That would certainly be true in the Blackhat community.

- A more formal definition would be "a security hole or an instance of taking advantage of a security hole".

- We can define an exploit as any tool or act that is used to compromise a system or network. These would include the following:

    - Gaining access to a restricted system
    - Bypassing or weakening network defenses to gain access
    - Taking a system offline (denial of service)
    - Acquiring sensitive or restricted information

- Any networks defenses are only as strong as its weakest link. Therefore it is important to examine and understand all the ways in which a system can be exploited.

- For an exploit to be present, there has to be a weakness that can be compromised. If there are no weaknesses, there is nothing to exploit.

- As a corollary then, as we minimize the number of possible exploits (increased security and defenses), the number of weaknesses is reduced or eliminated.

- It is also important to understand however that, in the process of minimizing the number exploits, the functionality of the system may also severely curtailed.

- Therefore, when designing security for a network, it is critical that we minimize the risk while reducing the impact that its deployment will have on overall functionality.

- There are many ways in which an attacker can gain access to or exploit a system. The method notwithstanding, there are some basic steps that are usually followed:

    1. Passive reconnaissance.

    2. Active reconnaissance (scanning).

    3. Exploiting the system:

        - Gaining access using Operating system attacks
        - Gaining access using Application-level attacks
        - Scripts and sample program attacks
        - Attacks on misconfigured systems
        - Elevating of privileges
        - Denial of Service

    4. Uploading programs.

    5. Downloading Data.

    6. Maintaining access using Backdoors and Trojans

    7. Covering tracks.

- The attacker would first seek any general information about the system. This consists of information like the domain name and any servers or systems the company might have.

- After all of the passive information has been gathered, active reconnaissance begins. This is where the attacker tries to find out as much information about the systems, without alerting any of monitoring systems.

- At this point the attacker has information such as IP addresses, open ports, operating system and version, and so on.

- Next, the attacker would step through each of the attack areas: operating system, applications, scripts, and misconfigured systems.

- For each item, an attacker tries an attack; if unsuccessful, he tries to gather more information about the component.

- After all the information has been gathered for an item, an attacker moves on to the next item.

- After an attack has been successful and access has been gained, the attacker then uploads any necessary programs, preserves access by installing Trojan horses, and finally cleans up the system to hide the attack.

Passive Reconnaissance

- In order to exploit a system, an attacker must have some general information; otherwise, she does not know what to attack.

- Passive information gathering is not always useful by itself, but it is a prerequisite to performing the other steps.

- In one case, I was gathering information to perform an authorized penetration test for a company.

- One of the most popular types of passive attacks is packet sniffing. This is a tool that can be installed on a subnet and capturing and examining all (or selectively) the packet traffic on that subnet.

- This can yield a lot of useful information for an attacker. For example, if an attacker is looking for a specific piece of information, he might simply write a filter that will search for the required pattern in the packet traffic.

- An example of this is sniffing passwords. There are programs that attackers can run from a workstation that looks for Win32 authentication packets. When it finds one, it extracts the encrypted password and saves it.

- An attacker can then use a password cracker to get the plain text password. To get a single password, this might seem like a lot of work.

- In a typical business environment most people will log on to the network between 0700 and 1000 hrs so hundreds of passwords can be gathered in a relatively short time period.

Active Reconnaissance

- This phase consists of probing the network to obtain additional information such as:
    - Hosts that are accessible
    - Locations of routers and firewalls
    - Operating systems running on key systems
    - Ports that are open
    - Services that are running
    - Versions of applications that are running
- The more information that an attacker can gain at this stage, the easier it will be to select the tools and method for the attack.
- This is an iterative process, where an attacker gathers some information, conducts a few covert tests and gathers more information until she gains access.
- It must be understood that as an attacker performs additional active reconnaissance, his chances of detection increase because he is actively performing some action against the company.
- The key to detecting this activity is to have some form of monitoring and logging in place to detect and record active reconnaissance.
- This activity has to be detected and blocked in its early stages, because, in a lot of cases, if the attacker is not blocked here, the chances of detecting her later decrease significantly.
- The goal of the security analyst is to deploy defense measures within a network and make it so difficult for attackers to gain access that they give up before they get in.
- The above statement is generally true for an opportunistic attacker who scans the Internet looking for any easy target.
- In cases of corporate espionage, where an attacker has targeted a specific site, some security will make the attacker's job more difficult, but will not necessarily stop her.
- In this situation, hopefully the extra security will make it so difficult that the attack will be detected before she gains access and stopped before any damage is done.

Exploiting the System

- When most people think about exploiting a system, they only think about **gaining access**. This is a common misapprehension because there are actually two other areas: **elevation of privileges** and **denial of services**.

- All three are useful to the attacker depending on the type of attack he wants to launch. They can also be used in conjunction with each other.

- For example, an attacker might be able to compromise a user's account to gain access to the system, but because he does not have root access, he cannot copy a sensitive file.

- At this point, the attacker would have to run an elevation of privileges attack to increase his security level so that he can access the appropriate files.

- It is also important to note that an attacker can exploit a system to use it as a launching pad for attacks against other networks.

- This is why system break-ins are not always noticed, because attackers are not out to do direct harm or steal information.

- In these cases, a company's valuable resources are being used and, technically, that company's systems are hacking into other networks.

- This is called a downstream liability problem. This can have substantial legal implications for a company, especially if it can be proven that the company that was used to launch another attack was negligent in its network defenses.

Gaining Access using Operating system attacks

- The more services a system offers, the more ports will be open and therefore more points of access and vice versa.
- Based on this one would expect that a default install of an operating system would have the least number of services running and ports open. This way, you control the points of compromise in a system.
- However, in reality, the opposite is true. The default install of most operating systems has large numbers of services running and ports open.
- The reason most manufacturers do this is simple, cost of support. They want a consumer of their product to be able to install and configure a system with the least amount of effort and trouble.
- Every time a consumer has a problem with their product they have to call for support, which costs the company large amounts of revenue.
- Fewer calls mean a smaller number of technical support staff, and lower costs. In addition, fewer support calls means that the user experiences less frustration, which increases satisfaction with the product.
- To further compound the problems, once the operating system is installed, most administrators are simply negligent or too busy to apply patches and updates.
- This leaves a company with outdated operating systems, which have a large number of vulnerabilities. A perfect opportunity for exploits.

Gaining Access using Application-level attacks

- Application-level attacks take advantage of the numerous security vulnerabilities found in most of today's application software.
- The programming development cycle for many applications leaves a lot to be desired in terms of security.
- The main problem with most software that is currently being developed is that the programmers and testers are under very tight deadlines to release a product.
- This usually results in a testing phase that is far less thorough than it should have been.
- This situation is further exacerbated by the ever-increasing functionality and complexity in applications so that even if there were more time to test, the chances of testing every feature would still be small.
- Poor or nonexistent error-checking accounts for a large number of security holes found in today's programs. Buffer overflows are just one example of this problem.
- In a complex operating system such as UNIX, extraneous scripts are responsible for a large number of entries and exploits.
- When the core operating system or application is installed, the manufacturers distribute sample files and scripts so that the owner of the system can better understand how the system works and can use the scripts to develop new applications.
- From a developer's standpoint, this is extremely helpful. Being able to use sample source code as a template helps increase the development time significantly.
- One of the main areas where there are a lot of sample scripts is in web development. The earlier versions of Apache web server and some web browsers came with several scripts and most of them had vulnerabilities.
- Also, a lot of the new scripting tools that come with web browsers enable developers with minimal programming knowledge to develop applications in a relatively short period of time.
- In these cases, the applications work, but behind the scenes there is usually a lot of extraneous code and poor error-checking, creating an open door for attackers.
- Active Server Pages (ASPS) are a perfect example of this. Much of the early development that occurred with ASPs left open backdoors that attackers are now exploiting.
- An example is the default web site that ships with IIS. It has the remote administration tools available from the main page. These tools can be used by an attacker to compromise a system.

Misconfiguration Attacks

- These attacks are directed towards systems that are not configured correctly. Often this happens when an administrator is working on a system and is not sure how to set it up, so she tries a variety of options until things start working.
- The problem with this is that she never goes back to figure out what made it work and to clean up the extraneous work that was done. This many times results in services and scripts being left enabled and leaving the system vulnerable.
- It is very important to remove any unneeded services or software. This way, the only things left on a system are the core components needed and you can concentrate **on** securing those.
- Misconfiguration is one area that is under your control as an administrator and so make sure you spend the time necessary and do it right.

Elevating Privileges

- The ultimate goal of an attacker is to gain either root or administrator access to a system.
- In some cases, an attacker can directly acquire this access. In other cases, an attacker has to gain a minimal amount of access and then elevate that to full access.
- For example, an attacker might acquire guest access and then use that access to gain additional information.
- After the additional information has been gained, the attacker uses this knowledge to increase his access to root or administrator access.

Denial of Service

- Denial of Service attacks are attacks that deny legitimate users access to a resource. These can range from blocking users from going to a particular web site to disabling accounts so that users cannot log on to a network.
- Unfortunately, these attacks are fairly easy to perform on the Internet because they require no prior access.
- Any system that is connected to the Internet is vulnerable to a Denial of Service attack.
- Tools for performing these types of attacks are readily available and very easy to use.

Uploading Programs

- After an attacker has gained access, she usually performs some set of actions on the server.
- There are few cases where an attacker gains access just for the sake of gaining access.
- Most often, she either uploads or downloads files or programs to or from the system.
- If an attacker is looking to steal information, after access is gained, the goal is to download information as covertly as possible and exit the system.
- In most cases, the attacker will upload programs to the system and use these to either increase access, compromise other systems on the network, or upload tools that will be used to compromise other systems.
- To cause damage or acquire information, an attacker must successfully break into a site and retrieve the necessary information.
- An added benefit for the attacker is that it is much harder to trace the attack. If an attacker is launching an attack from Company A and she covers her tracks and breaks into Company B, Company B can see only that Company A attacked it.
- The fact that an attacker was able to break into Company A in the first place usually means that Company A has weak security, which makes it extremely difficult to trace it back to the originator.

Downloading Data

- In the case of corporate espionage, an attacker is after information. This information can range from data about a new research and development product, a customer list, or future direction of the company.
- In all of these cases, the attacker wants access to download data to another location. After the data is downloaded to another location, an attacker can perform whatever analysis he needs to on the information.
- The thing to keep in mind about this type of attack is that if you do not detect the attacker when he is downloading the data, you have no chance of stopping the attack.
- After the data has been downloaded, the remainder of the attack is done offline.

Keeping Access

- After an attacker gains access to a system, she will put in a back door so that she can return whenever she wants.
- In most cases, an attacker has gained root equivalent access and she can do whatever she wants on the system, and usually the first order of business is to install a back door.
- As mentioned earlier, another reason attackers want to maintain access is to use those machines as a staging area to launch attacks against other companies.
- A back door can be as simple as adding an account to the system. This is simple, but if the company reviews its active accounts, it has a high chance of detecting it.
- However, if it is a system with thousands of users, chances are that no one will notice.

Covering Tracks

- An attacker's last step is to cover his tracks. At the most basic level this means cleaning up the log files.
- The log files keep a record of all user activities including logins, services accessed, times, etc.
- From an attackers standpoint, this is a not a good thing. So, to cover his tracks, he first finds out where the log file is and clears out the entries that are related to the attack.
- Note that an experienced attacker will not delete the entire contents of the log file to ensure that the evidence of the compromise is destroyed.
- There are two major drawbacks to total deletion. First, empty log files raise immediate suspicion that something is wrong.
- Second, most systems put an entry in the log file indicating that the file has been cleared.
- That is why it is so important to send logging to a remote machine and ideally have the log information go to a write-only medium.
- Another common hacker technique is to turn off logging as soon as she gains access to a machine.
- This way, no one is the wiser about what has been done. This requires additional expertise, but it is extremely effective.
- The thing to remember is that if logging is done correctly, even if an attacker turns off logging, the system still records the fact that she entered the system, where she entered, and other useful information.

- If an attacker modifies or overwrites files, part of his cleaning-up process is to make sure that the changed files do not raise suspicion.

- Most files have dates of when they were last accessed and the size of the file. There are programs that, when run, raise flags if information has been changed.

- To overcome this, an attacker can go in and trick the system. Even though the file has been modified and the size has changed, he can go into the properties of the files and set them back to their previous settings, which make it much harder to detect.

- It is highly recommended that systems run programs like tripwire that maintains a database of checksums for key system programs.

- A checksum is a calculation performed on the file, and two checksums can only be the same if the files are identical.

- This means that even if an attacker goes in and tries to cover his tracks, because the file changed, the checksums should be different