Exploitation

Contents

Purpose

The exploitation phase of a penetration test focuses solely on establishing access to a system or resource by bypassing security restrictions. If the prior phase, vulnerability analysis was performed properly, this phase should be well planned and a precision strike.. The main focus is to identify the main entry point into the organization and to identify high value target assets.

If the vulnerability analysis phase was properly completed, a high value target list should have been complied. Ultimately the attack vector should take into consideration the success probability and highest impact on the organization.

Countermeasures

Countermeasures are defined as preventative technology or controls that hinder the ability to successfully complete an exploit avenue. This technology could be a Host Based Intrusion Prevention System, Security Guard, Web Application Firewall, or other preventative methods. When performing an exploit, several factors should be taken into consideration. In the event of a preventative technology, a circumvention technique should be considered. In circumstances when this is not possible, alternative exploit methods should be considered.

Overall, the purpose is to remain stealth when attacking the organization, if alarms are tripped the level of the assessment could be diminished. If at all possible, the countermeasures should be enumerated prior to triggering the exploit. This could be done through doing dry runs of the attack or enumerating the technology.

Anti-Virus

Anti-virus is a technology aimed at preventing malicious software from being deployed on the system. As a penetration tester we should be able to identify these types of anti-virus technologies and be able to protect against them. Anti-virus is a small subset of all of the different preventative measures that can be in place, for example host-based intrusion prevention systems, web application firewalls, and other preventative technologies.

Encoding

Encoding is the method of obfuscating data in a way that makes the deployed piece of code not appear the same. With encoding, the obfuscation occurs usually by scrambling the information and re-arranging in order to hide the fact of what the application is actually doing.

Packing

Packing is similar to encoding in a sense in that it attempts to re-arrange data to compress the application or "pack" it. The hopes of this is that the executable or piece of code being delivered is obfuscated in a manner that it won't be picked up by anti-virus technologies.

Encrypting

Encrypting, like Encoding and Packing is another method of manipulating the intended runnable code such that it is not recognizable or available for inspection. Only after decrypting in in-memory (with methods similar to packing) the actual code is exposed for the first time - hopefully after security mechanisms have allowed it through and it is executed immediately after it is decrypted.

Whitelist Bypass

Whitelisting technologies leveraged a trusted model for applications that have been seen on a given system at a time. The technology takes a baseline of the system and identifies what is normal to be run on the system versus what is something foreign. The penetration tester should be able to circumvent whitelist technologies. One of the most common methods is through direct memory access. Whitelisting does not have the capbility of monitoring memory real time and if a memory resident program is running and not touching disk, it can run without being detect by the given technology.

Process Injection

Process injection is simply the method to inject into an already running process. By injecting into a process, the information of the application can be hidden within a process that would normally be trusted in nature. It's very difficult for preventative measure technology to inspect running processes and can almost always hide in a different process that the application would think is a trusted one.

Purely Memory Resident

Memory resident attacks are generally the most preferred as most technologies do not inspect memory. As an attacker, finding a way to live in memory purely would be most desirable. When writing to disk, most applications will conduct scans, baselines, and other identifications of potentially maliicous software. The ability to be detected when writing to disk becomes significantly greater.

Human

When performing exploitation, it is not always the best route to go through a direct exploit or through an application flaw. Sometimes the human element may be a better way to attack an organization. It's important to understand the right attack avenue and make sure that the method we are leveraging is the best route to take.

Data Execution Prevention (DEP)

When performing exploitation, many preventative measures canc ome into play. Data Execution Prevention is a defensive measure implemented into most operating systems and prevents execute permission when an overwrite in memory has occurred. The thought process behind DEP is to stop an attacker in rewriting memory and then executing that code. There are mulitple methods to bypass data execution prevention and discussed later in the the exploitation phase of PTES.

Address Space Layout Randomization

During a buffer overflow vulnerability (or that of anything where we control memory), memory addresses are hardcoded in order redirect execution flow to our shellcode. In the event of ASLR, certain bytes are randomized in order to prevent an attacker from predicting where he/she can always go to in order to execute shellcode.

Web Application Firewall (WAF)

Web application firewalls are a technology that sits inline with an application in order to protect against web-based application attacks. Web application firewalls attempt to identify potentially dangerous or malforms attacked towards a given web applciation and prevent them. There are a

number of bypass techniques for web application firewalls and should be tested during the penetration test.

Evasion

Evasion is the technique used in order to escape detection during a penetration test. This could be circumventing a camera system as to not be seen by a guard, obfuscating your payloads to evade Intrusion Detection Systems (IDS) or Intrusion Prevention Systems (IPS) or encoding requests/responses to circumvent web application firewalls. Overall, the need to identify a low risk scenario for evading a technology or person should be formulated prior to the exploit.

Precision Strike

The main focus of a penetration test is to simulate an attacker in order to represent a simulated attack against the organization. The value brought through a penetration test is generally not through smash and grab techniques where the attacks are noisy in nature and in an attempt to try every exploit. This approach may be particularly useful at the end of a penetration test to gauge the level of incident response from the organization, but in most cases the exploitation phase is a accumulation of specific research on the target.

Customized Exploitation Avenue

Every attack will typically not be the same in how the exploitation avenue occurs. In order to be successful in this phase, the attack should be tailored and customized based on the scenario. For example, if a wireless penetration test is occurred, and a specific technology is in use, these need to be identified and attacked based on what technologies are in place. Having a clear understanding of each scenario and the applicability of an exploit is one of the most important aspects of this phase of the penetration test.

Tailored Exploits

In a number of occasions the exploits that are public on the Internet may need some work in order to successfully complete. In most cases, if an exploit is designed for Windows XP SP2, specific modifications to the exploit will be required in order for the attack to be successful via Windows XP SP3. The penetration tester should have the knowledge in place to be able to customize an exploit and the ability to change on the fly in order to successfully complete the attack.

Exploit Customization

In the event of an attack, it is often required to simulate the victims infrastructure in order to ensure that the exploitation phase will be successful. The techniques leveraged in the information gathering phase can always help assist in that however, having a working infrastructure and systems in place will make the expliotation phase much easier. In the event of a tailored exploit, the penetration tester should be able to customize already public exploits in order to successfully attack a system. A common theme for exploits is to target specific versions of operating systems or applications. The reason for this is due to memory addresses changing based on service packs, and/or new versions of the operating system. The tester should be able to customize these exploits to successfully deploy to different operating systems and successfully compromise the system.

Zero-Day Angle

In most cases, the zero-day angle is often a last resort for most penetration testers. This type of attack often represents a highly advanced organization that can handle a focused attack against the organization through normal attack methods. In certain scenarios research may be conducted in order to reverse engineer, fuzz, or perform advanced discovery of vulnerabilities that have not been discovered. In the event this type of attack is applicable, ensure that the environment to the best of the attackers knowledge is reproduced to include countermeasure technology.

In order for zero-day exploits to be successful (or any exploit for that matter), having the same operating system, patches, and countermeasures is highly important on success. Sometimes this information may not be available based on the level of access or enumeration that has occurred.

Fuzzing

Fuzzing is the ability to recreate a protocol or application and attempt to send data at the application in hopes of identification of a vulnerability. Often times the hopes of a fuzzer is to identify a crash in an application and craft a specific exploit out of it. In the case of fuzzing, the attacker is attempting to create a specific vulnerability out of something that hasn't been discovered before. As part of a penetration test, if no avenues are identified during the engagement, or the engagement calls for zero-day research; fuzzing techniques should be leveraged in order to identify potentially vulnerable exposures.

Source Code Analysis

Other avenues that a penetration tester has available is if the source code is available or open-source. If the tester has the ability to look at the source code and identify flaws within the application, zero day exposures can also be identified through these methods.

Types of Exploits

There are several types of exploits that can be identified during a penetration test that could be classified as a zero-day. Some are listed in this section.

Buffer Overflows

Buffer overflows occur due to improper coding techniques. Specifically this usually occurs when a program writes data to a buffer and then overruns the buffer's boundary and begins to overwrite portions of memory. In buffer overflow exploits the attackers goal is to control a crash and gain code execution on the given system. In a buffer overflow exploit, one of the more common techniques is to overwrite a given register and "jump" to the shellcode.

SEH Overwrites

SEH overwrites occur when the structured exception handler begins to gracefully close an application. The attacker can manipulate how SEH works, overwrite the base address of the SEH handler and gain control of execution flow through the SEH. This is a common attack leveraged with buffer overflow vulnerability and applications that have been complied with SEH.

Return Oriented Programming

Return Oriented Programming (ROP) is a technique used during a portion where the user has control of execution flow however data execution prevention (DEP) or other precluding defense mechanisms may be in place. In the situation where DEP is enabled, the attacker does not have direct access to execute specific assembly instructions, therefor the attacker buildsa ROP gadget in order to prep certain Windows API calls or techniques to disable DEP or circumvent DEP. A common method is leveraging the WriteProcessMemory call to copy data from the stack into a writable memory space that can then be executed.

Traffic Analysis

Traffic analysis is the technique of identifying what type of information is being sent and the ability to understand and manipulate that traffic. A penetration tester should be able to understand how a protocol works and how it can be manipulated in order to leverage an attack.

Physical Access

Physical access during a penetration test can be a viable attack method for attempting to circumvent physical security controls and gain unauthorized access. During a penetration test, the assessor should be able to identify potentially flawed physical security controls and attempt to gain access to the facility if within scope.

Human Angle

During a physical penetration test, some of the most obvious ways would be to social-engineer your way into the facility and gain access. This requires significant knowledge of how the organization performs business, and everything you learned from the intelligence gathering phase.

PC Access

If physical access is granted to a PC, the penetration tester should be able to attack the PC and gain access through multiple methods that would allow access to the system.

Proximity Access (WiFi)

Wireless communications are an avenue for attacks to gain access through RF type communications. The penetration tester should view the FCC radio frequency list to see if the target has registered spectrum frequencies in use.

WiFi Attacks

Regardless of protocol, there are a number of attacks available for WEP, WPA, WPA2, EAP-FAST, EAP-LEAP, and other avenues. The attacker should be familiar with the various encryption protocols and standards and be able to effectively test the implementation around the controls put in place.

Attacking the User

Leveraging rogue access points in order to attack the victim is often a beneficial and a viable attack method. Leveraging a rogue access point to entice victims in order to leverage exploits or steal sensitive information should be performed during a wireless assessment. There are several common

techniques in use of this, but most commonly the attacker would setup a wireless access point with the same name or an enticing name in order for the victim to connect.

Example Avenues of Attack

In any scenario, the attacks should consist based on the scenario that is within scope of the engagement. Below is a list of several attack avenues to consider based on scenario but is by no means a comprehensive list.

Web Application Attacks Social-Engineering Physical Attack Avenues Memory Based Exploits (i.e. buffer/heap overflows, memory corruptions, use-after-free). Man in the Middle VLAN Hopping USB/Flash Drive deployment Reverse Engineering Zero-Day Angle Attacking the user Encryption Cracking Graphics Processing Unit (GPU) Cracking Traffic Analysis Firewire Routing protocols Phishing with Pretexting Employee Impersonation

Again, these examples are only basic avenues for attack based on the scenario you are performing for the organization. The value from a penetration test comes from creativity and the ability to identify exposures and exploit them in a precise manner.

Overall Objective

In the pre-engagement interaction phase with the customer, a clear definition of the overall objectives of the penetration test should have been communicated. In the case of the exploitation phase, the biggest challenge is identifying the least path of resistance into the organization without detection and having the most impact on the organizations ability to generate revenue.

By performing the prior phases properly, a clear understanding of how the organization functions and makes money should be relatively understood. From the exploitation phase and into the post-exploitation phase, the attack vectors should rely solely on the mission of circumventing security controls in order to represent how the organization can suffer substantial losses through a targeted attack against the organization.
