Windows Privilege Escalation Techniques and How to Mitigate Them

There are many privilege escalation methods in Windows operating systems. Here is a brief review of three common methods and how you can prevent them.

Access Token Manipulation

Attack description:

Windows uses access tokens to determine the owners of running processes. When a process tries to perform a task that requires privileges, the system checks who owns the process and to see if they have sufficient permissions. Access token manipulation involves fooling the system into believing that the running process belongs to someone other than the user who started the process, granting the process the permissions of the other user.

In this method, an adversary has a username and password, but the user is not logged.

Mitigation

There is no way to disable access tokens in Windows. However, to perform this technique an attacker must already have administrative-level access. The best way to prevent the attack is to assign administrative rights in line with the least-privilege principle, regularly review administrative accounts and revoke them if access is no longer needed. Also, monitor privileged accounts for any sign of anomalous behaviour.

Bypass User Account Control

Attack description

The Windows user account control (UAC) mechanism creates a distinction between regular users and administrators. It limits all applications to standard user permissions unless specifically authorized by an administrator, to prevent malware from compromising the operating system. However, if UAC protection is not at the highest level, some Windows programs can escalate privileges, or execute COM objects with administrative privileges.

Mitigation

Review IT systems and ensure UAC protection is set to the highest level, or if this is not possible, apply other security measures. Regularly review which accounts are a local administrator group on sensitive systems and remove regular users who should not have administrative rights.

DLL Search Order Hijacking

Attack description

Attackers can perform “DLL preloading”. This involves planting a malicious DLL with the same name as a legitimate DLL, in a location which is searched by the system before the legitimate DLL. Often this will be the current working directory, or in some cases attackers may remotely set the working directory to an external file volume. The system finds the DLL in the working folder, thinking it is the legitimate DLL, and executes it.

Mitigation

Here are several ways to prevent a DLL search order hijack:

- Disallow loading of remote DLLs
- Enable Safe DLL Search Mode to force search for system DLLs in directories with greater restrictions.
- Use auditing tools such as PowerSploit to detect DLL search order hijacking vulnerabilities and correct them.
- Identify and block software executed through search order hijacking, using whitelisting tools like AppLocker.

Linux Privilege Escalation

What Is Enumeration?

In Linux systems, attackers use a process called “enumeration” to identify weaknesses that may allow privilege escalation. Enumeration involves:

- Using Google searches, port scanning and direct interaction with a system to learn more about it and see how it responds to inputs.
- Seeing if compilers, or high-level programming languages like Perl or Python, are available, which can allow an attacker to run exploit code.
- Identifying software components, such as web servers and their versions.
- Retrieving data from key system directories such as /etc, /proc, ipconfig, lsof, netstat and uname.

Attackers use automated tools to perform enumeration on Linux systems. You should also use the same tools to pre-empt an attack, by scanning your own system, identifying weaknesses, and addressing them.

Below are two specific techniques for escalating privilege on Linux and how to mitigate them.

Kernel Exploit

Attack description

From time to time, vulnerabilities are discovered in the Linux kernel. Attackers can exploit these vulnerabilities to gain root access to a Linux system, and once the system is infected with the exploit, there is no way to defend against it.

Attackers go through the following steps:

1. Learn about the vulnerabilities
2. Develop or acquire exploit code
3. Transfer the exploit onto the target
4. Execute the exploit on the target

Mitigation

Follow security reports and promptly install Linux updates and patches. Restrict or remove programs that enable file transfers, such as FTP, SCP, or curl, or

restrict them to specific users or IPs. This can prevent transfer of an exploit onto a target device. Remove or restrict access to compilers, such as GCC, to prevent exploits from executing. You should also limit which folders are writable or executable.

Exploiting SUDO Rights

Attack description

SUDO is a Linux program that lets users run programs with the security privileges of another user. Older versions would run as the superuser (SU) by default. Attackers can try to compromise a user who has SUDO access to a system, and if successful, they gain root privileges.

A common scenario is administrators granting access to some users to perform supposedly harmless SUDO commands, such as ‘find’. However, the ‘find’ command container parameters that enable command execution, and so if attackers compromise that user’s account, they can execute commands with root privileges.

Mitigation

Never give SUDO rights to the programming language compiler, interpreter or editors, including vi, more, less, nmap, perl, ruby, python, gdb. Do not give sudo rights to any program that enables running a shell. And severely limit SUDO access using the least-privilege principle.