CRASHTEST **SECURITY**

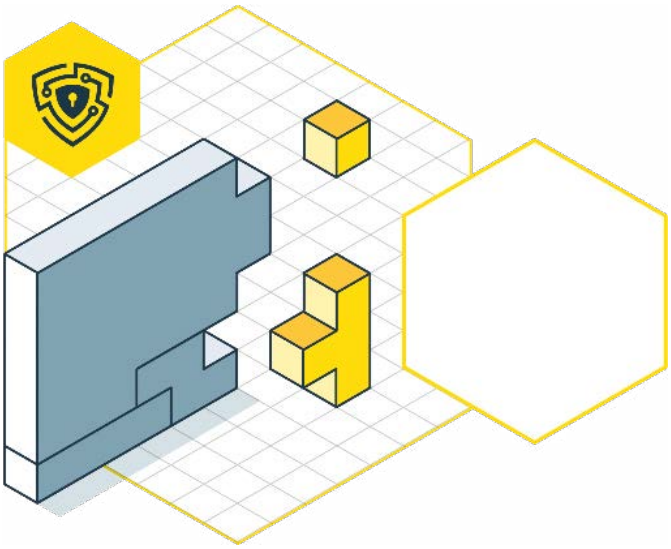
2022

GUIDE FOR PRIVILEGE ESCALATION

WHAT ARE THE STEPS TO KEEP YOUR WEB APP OR API SAFE FROM SUCH VULNERABILITY

GUIDE FOR THE PRIVILEGE ESCALATION

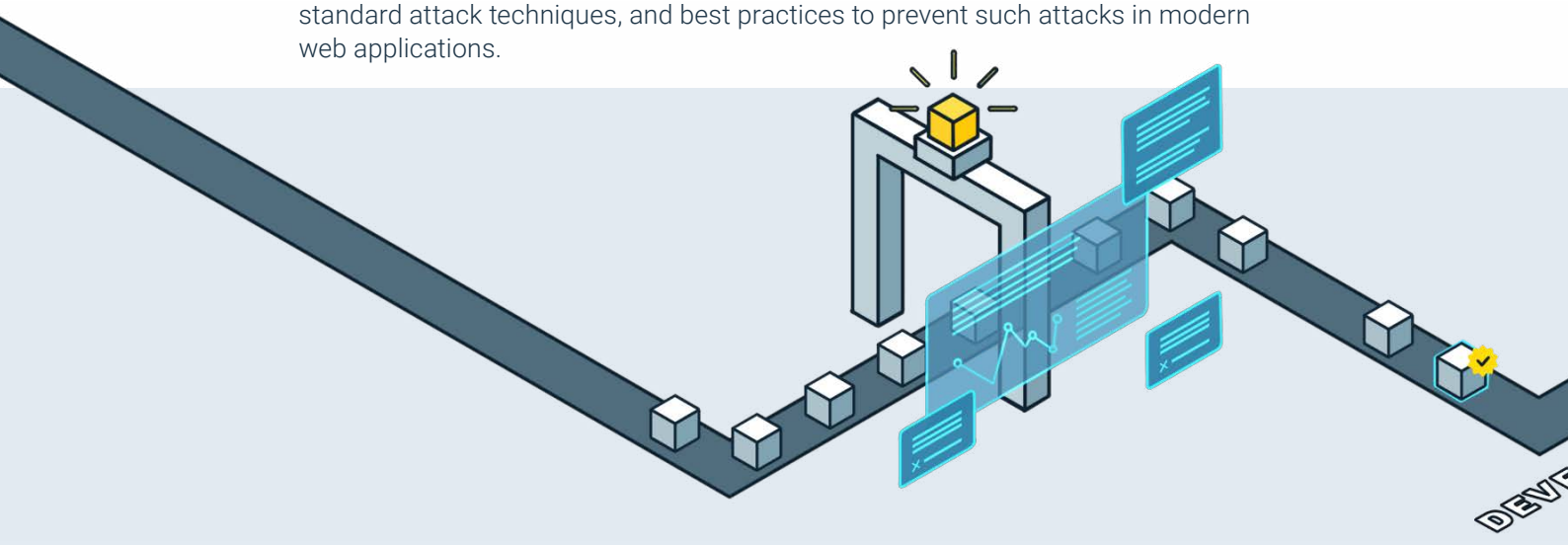
Table of Contents



INTRODUCTION TO THIS GUIDE

Most modern applications follow a multi-tenant architecture designed to be used by multiple users with varying access privileges. A cyber attack usually begins by compromising one of the standard user's access to gain entry into the system. The next step usually involves elevating access to privileged accounts and resources kept away from a standard user. This form of a security breach is commonly known as a privilege escalation that facilitates illicit access to escalated rights and permissions beyond what the user is entitled to.

Privilege escalation is one of the most commonly leveraged mechanisms in a modern cyber attack chain that results in unauthorized access to an entire system. This guide delves into how privilege escalation works, its various severity levels, standard attack techniques, and best practices to prevent such attacks in modern web applications.



WHAT IS PRIVILEGE ESCALATION?

In a privilege escalation attack, hackers leverage vulnerabilities in access controls and resource restrictions to override the permissions and limitations of a target user account for gaining elevated access. These attacks are usually aimed at obtaining administrative privileges and utilizing them to manipulate the system's security settings by further extending the scope of the initial attack. Although most attacks rely on compromised unprivileged user accounts, attackers can also escalate privileges by exploiting other configurations and system bugs.

Some common threat vectors for privilege escalation attacks include:

Known Misconfigurations

In the absence of robust security practices during the fundamental phases of the application design, inherent misconfigurations are one of the most common exploits for malicious activities. Most privilege escalation attacks target misconfigurations in user accounts with poor or default security settings. Other misconfigurations that can be used for privilege escalation include:

- Exposing storage buckets to public networks with no authentication
- Using default credentials for root and admin accounts
- Undocumented administrative backdoors that attackers can discover
- Poorly documented configuration changes
- Improper configuration of the latest security patch in an upgraded component
- Error messages revealing stack traces or other sensitive information
- Installation and activation of unnecessary features such as ports, accounts, and services
- Missing validation for user input
- Insecure or missing security directives and headers in server responses

Privilege Vulnerabilities

Privilege vulnerabilities refer to those system flaws that allow users to adjust their permissions once they have been authenticated to the system. Examples of privilege escalation vulnerabilities in Windows and Linux systems include:

- Attackers utilize **access token manipulation** to trick the server into believing that a malicious process belongs to a legitimate user and gaining explicit permission for the malicious actor as that of the user. To get the access token, attackers orchestrate various techniques such as access token duplication, creating a login session using the LogonUser function, or creating a new process using a duplicated token.

- **Bypassing the user account control (UAC) mechanism** helps obtain elevated privileges by abusing the distinction between administrative and standard users.

- Attackers commonly **hack the DLL search order** by planting an adversarial DLL with the same name as the legitimate DLL, but in a location that the system will search before the legitimate DLL. This location is either at an upper-level folder within the working directory or a remote directory pointing to an external file volume. This results in the application finding the malicious DLL and executing it, believing it to be a legitimate DLL.

- An attacker connects to the host and employs the mechanism of **enumeration** to gather information about usernames, machine names, services, and networks. This allows them to discover potential attack vectors to access additional resources deep into the system.

- Attackers exploit known **Linux Kernel vulnerabilities** to gain root-level access to the system for executing attacks at the root privilege level, making it impractical for the system to defend.

- Attackers typically target users with SUDO system access to **exploit** their **SUDO rights** to gain root access for executing commands.

Credential Exploitation

Flaws in the authentication layer often equip malicious users with knowledge of the account names, user IDs, and passwords needed for login attempts. Over the years, attackers have developed several sophisticated ways of acquiring user credentials, such as:

- **Password exposure** through hard-coded passwords or insecure credential databases
- Utilizing **brute force attacks** to guess passwords using known and common combinations
- Using devices such as keyloggers and cameras to **shoulder surf** actions of authorized and privileged users
- Utilize **dictionary attacks** and **automated routines** to combine lists of common words that can be used as access credentials
- Leverage algorithms to encrypt and hash passwords while utilizing **Rainbow table attacks** to reconstruct hashes into original passwords
- Using the **Credential stuffing** technique to gain the lists of exposed usernames and passwords from previous exploits. This works well as users often reuse passwords across services
- When a user's device gets compromised, attackers commonly exploit the weaknesses of the process in generating, transmitting, and storing a new password when requested for an account password reset.

Malware

Malware is one of the most infamous malicious software that runs as an operational process with permissions of the compromised user account who executed it. Attackers can deploy malware at the standard user level and later escalate the privileges to the administrative level, extending the radius of their attack to an entire ecosystem. Malware that can be used for privilege escalation includes:

- **Rootkits** - These are malicious processes running with root privileges, giving the threat actor complete control over the Operating Systems.
- **Bots** - They are automated programs used to spread arbitrary code and malware and perform other malicious actions against the compromised assets.
- **Spyware** - These refer to the malicious software that performs reconnaissance and surveillance on target devices/users. Some examples of Spyware are keyboard loggers, webcam hacking software, microphone bugs, etc.

Social Engineering

These techniques manipulate users into violating security controls and exposing sensitive information. Since it preys on human weaknesses, social engineering is one of the most effective techniques for privilege escalation. Some common social engineering attacks used for privilege escalation include:

- **Phishing** - Phishing tricks the user into clicking a malicious link or attachment in a legitimate-looking message. Clicking the link or attachment typically initiates the deployment of malware or other compromises used to obtain illegal access.
- **Whaling** - Whaling is a form of phishing attack aimed at high-ranking personnel of an organization. This typically involves using fake time-sensitive opportunities and emergencies to pressurize executives into clicking malicious links to expose credentials for high-level network access.
- **Baiting** - The baiting technique lures the victim user into exposing sensitive data by being promised a gift.
- **Scareware** - The attacker uses alarming audio and flashy graphics through pop-up windows in a scareware attack. The intention is to falsely alarm the user of an impending upgrade/virus that needs to be purged. Clicking on the link sends the user to an antivirus purchase website where the scammer may attempt to steal credit card or other account information.

IMPACT OF PRIVILEGE ESCALATION ATTACK ON AN ORGANIZATION

Privilege escalation remains a critical concern for an organization‘s web application security. The attacks are no longer restricted to the network periphery but intrude inside the organization‘s database to gain access to sensitive customer/organization internal data. Privilege escalation is often the first stage of a crucial complicated attack. The ultimate goal of the attack is always to gain elevated and unauthorized access to an organization‘s critical resources and compromise security. Impacts might range from accessing confidential data, installing malware or malicious code, and/or completely hijacking the organization‘s systems.

TYPES OF PRIVILEGE ESCALATION

Privilege escalation attacks are broadly categorized into two categories based on how the attacker extends the reach of its attack. These include:

HORIZONTAL PRIVILEGE ESCALATION

In a horizontal privilege escalation attack, the hacker gains the access rights of other entities with similar permissions. To do so, attackers typically target unprotected, lower-level user accounts with similar privileges but have access to different security contexts. By orchestrating such attacks, attackers can gain access to resources and features that are available only to the victim user.

Some vulnerabilities that facilitate horizontal privilege escalation include:

- User IDs controlled by HTTP request parameters in the URL
- Exposed globally unique identifiers (GUIDs)
- Data leakage in redirects that contain user IDs

VERTICAL PRIVILEGE ESCALATION

In a vertical privilege escalation attack, attackers leverage flaws in the system to upgrade the access rights of a compromised account that allows them to view and control resources they are currently not permitted to do. Vertical escalation, also known as **privilege elevation**, typically entails moving from low-level users to a higher form of privileged access, often administrative. This typically requires going through various intermediate steps to obtain privileged credentials. Some commonly used steps include: **obtaining credentials for privileged accounts, editing high-level scripts and executables, editing application source code,** and **abusing misconfigurations to gain access to these accounts**.

Vulnerabilities that are most susceptible to vertical privilege escalations include:

- Unprotected functionality
- Parameter-based access controls
- Platform configuration flaws leading to broken access control

WHAT IS THE SEVERITY LEVEL OF PRIVILEGE ESCALATION?

Privilege escalation is a broken access control flaw, which is ranked number one in the OWASP top ten list of vulnerabilities. By allowing a compromised user to act beyond their intended permissions, such vulnerabilities often lead to unauthorized access, affecting the integrity and availability of the application. With a privilege escalation attack, hackers can leak personal information belonging to the application's users, manipulate the data processed by the application or compromise the execution of processes by the webserver if the application processes financial or commercial data, horizontal privilege escalation attacks are mainly carried out for theft and fraud through account hijacking and data manipulation.

Privilege escalation vulnerability is considered highly exploitable (exploitability: two) since most threat actors are skilled at detecting and leveraging access control failures. At the same time, it is essential to note that both manual and automated techniques are available to detect such flaws within a system. When developing application code, lack of functional testing is considered one of the most common aberrations that make this vulnerability relatively common (prevalence: two).

Since a privilege escalation attack allows hackers to initiate administrative functions, the technical and business impacts vary from low to critical depending on the data and processes run by the application. Typical consequences include exposure of sensitive information, data alteration/destruction, malicious code execution, and service denial.

HOW TO IDENTIFY PRIVILEGE ESCALATION VULNERABILITIES WITH CRASHTEST SECURITY

Crashtest Security offers an automated vulnerability testing tool to help prevent privilege escalation by identifying and remediating access control flaws. The security suite includes a custom privilege escalation scanner that helps prevent attackers from gaining administrative rights to web applications. Crashtest Security's DAST tool also helps embed security in the initial phases of development to eliminate flaws in the application's peripheral such as open ports and APIs, that may lead to account takeovers.

The scanner outputs a privilege escalation report that lists attack vectors based on categories, severity, and proposed mitigation solutions. Crashtest Security also includes a port, OWASP, and HTTP header scanner to help track down specific flaws and misconfigurations commonly exploited to orchestrate privilege escalation attacks.

PRIVILEGE ESCALATION PREVENTION TECHNIQUES

Given a large number of exploitation techniques, preventing privilege escalation attacks requires multiple defense strategies like privileged access management and identity-centric patterns. Some robust approaches to prevent privilege escalation attacks include:

STRONG PASSWORD POLICIES

Developers and security professionals should enforce password policies that promote the creation of strong, secure passwords which cannot be compromised using social engineering techniques. Users should also be discouraged from reusing passwords across services that may get compromised in other data breaches. Additionally, more robust authentication controls such as Multi-Factor Authentication (MFA) and device-based sign-in restrict malicious players from accessing administrative accounts using stolen credentials.

ENFORCE THE PRINCIPLE OF LEAST PRIVILEGE

Every entity that interacts with the application should only be granted the privileges required to complete its task. The assignment of access rights should be controlled by the role and function of the entity rather than the identity. These principles confine processes to a small domain, reducing the blast radius of a successful attack. A policy must be enforced for developers to mandatorily declare and document the permissions required to access each resource while administrators should perform appropriate security checks before granting access. Assigning only the needed permissions makes it difficult for hackers to perform activities outside their intended scope.

SECURE MANAGEMENT OF PRIVILEGED ACCOUNTS

It is crucial to secure privilege accounts against exposure and unintended access. The security teams should catalog all privileged accounts along with the processes they execute. An application should have minimal privilege accounts with defined scopes. Additionally, security professionals should set up continuous logging and monitoring for these accounts to avoid any exploit. These accounts should also be constantly tested and hardened against potential threats and vulnerabilities that may lead to abuse of the privilege.

BEST PRACTICES IN PREVENTING PRIVILEGE ESCALATION

Some of the recommended practices to prevent privilege escalation include:

Keep Components Updated

Most privilege escalation attacks target known vulnerabilities that are readily published in the Common Vulnerability Enumeration database. Security teams should ensure the systems and applications that interact with the webserver are regularly patched with the latest updates to minimize the risk of hacking attacks. A trusted patch management system should also be integrated with appropriate software implementations to help applications not miss any critical security updates.

Eliminate File Transfer Functionality

Attackers mostly use file transfer functions such as FTP, get, and curl to download and execute malware. These tools and functions should be used when necessary with utmost precaution and should only be enabled for specific users, directories and applications.

Invest in Security Training and Awareness

As most attacks begin with social engineering techniques, the human factor is considered the weakest link in a cyberattack chain. All employees and third-party vendors should be educated on common cyber security risks and their roles in avoiding them.

Vulnerability Assessment and Mitigation

Automated scanners should be used to constantly scan, detect and identify flaws that could lead to privilege escalation. Identified vulnerabilities should be urgently addressed with patches, updates, and fixing misconfigurations to strengthen the web application's security posture.

BEST PRACTICES IN PREVENTING PRIVILEGE ESCALATION

Crashtest Security Suite helps reduce the attack surface by enforcing **continuous penetration testing** and **automated scanning** for broken access controls and security misconfigurations. The suite comes bundled with multiple automated scanners with simple integration for end-to-end cyber security hardening. Start your **free, fourteen-day trial** to see how Crashtest Security can help eliminate privilege vulnerabilities.

Start 2-Week Trial for Free

CRASHTEST **SECURITY**

WWW.CRASHTEST-SECURITY.COM