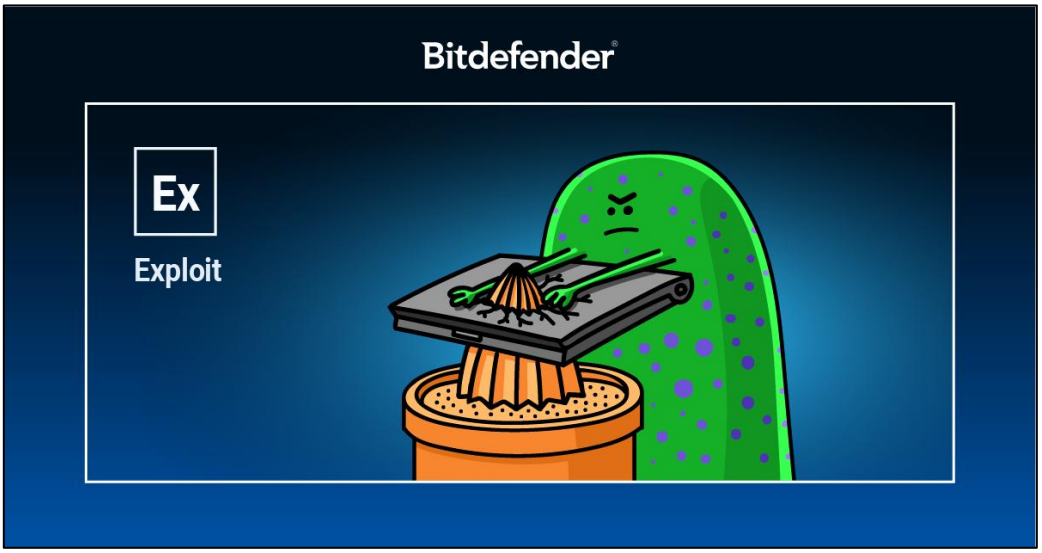
What is an Exploit? Exploit Prevention

This article explains what an Exploit is in cybersecurity. We'll also give you examples of exploits and how hackers can exploit vulnerabilities in applications, networks, operating systems, or hardware to attack a system. Additionally, we'll highlight tips to help you mitigate the risk of exploits and to protect against exploits.

What is an Exploit in cybersecurity?

An exploit is a piece of software, a chunk of data, or a sequence of commands that takes advantage of a bug or vulnerability in an application or a system to cause unintended or unanticipated behaviour to occur. The name comes from the English verb *to exploit*, meaning "to use something to one's own advantage". Basically, this means that the target of an attack suffers from a design flaw that allows hackers to create the means to access it and use it in their interest.



Users often ignore security warnings from Apple, Adobe, or Microsoft under the pretext that they don't have the time or patience to update software. This exposes them to more cyberattacks – some more sophisticated than others. When you don't update an outdated browser or plug-in, you automatically allow a hacker to take advantage of existing and even already discovered vulnerabilities. This gives the hacker full access to the data and software installed on your device. This cyber threat is no joke. Such an attack can have very serious consequences.

In a perfect world, applications would run flawlessly: no sudden crashes in the middle of your work, and no flaws in the way they have been designed. However, in real life, the complexity of software often leaves hackers room to scout for flaws and turn them against the user. They take months or even years to investigate the inner workings of highly popular software applications and to find ways to force them into behaving unexpectedly. When it is first discovered, such a vulnerability is called a zero-day exploit – an exploit that has not been seen before and for which the software vendor does not have a patch readily available.

The timeframe between the first use of the exploit and the release of a patch to fix it is called the “vulnerability window” and represents the period during which the user can be attacked without being able to fix the exploited flaw. On underground forums, zero-day exploits are sold for anywhere between $10,000 and $500,000, depending on the affected platform and its popularity on the market.

When a hacker “exploits” a device, it means that such a bug or software vulnerability has been weaponized (i.e. paired with malware) and it is actively pushed to the user via web pages or removable media.

Operating systems are not the only victims. These types of attacks target any software, hardware, or electronic device that can download files from the internet. Some of the most common targets are Microsoft Office, web browsers such as Internet Explorer, media players such as Adobe Flash Player, Adobe Reader, and unpatched versions of Oracle Java.

There are two types of exploits: local and remote. Local exploits are more sophisticated because they involve prior access to the system, while remote exploits manipulate the device without first requiring access to the system.

How do hackers exploit vulnerabilities to attack a system?

To detect vulnerabilities and exploit them, hackers must first gain access to the device. To do this, they use the simplest tool: social engineering. They will manipulate the user to open a suspicious email or attachment that loads specially crafted content into the vulnerable plug-in. Once rendered, the content usually causes the application to crash and inadvertently installs a malicious payload without user intervention.

Often, exploits are bundled into an exploit pack – a web application that probes the operating system, browser and browser plugins, looks for vulnerable applications and then pushes the app-specific content to the user.

It’s not difficult for a criminal to discover problems in a system. Protocol vulnerabilities are not immediately identified by vendors or security researchers, so by the time a patch is released, hackers may have already launched a zero-day exploit attack. Zero-day attacks are hard to combat and have increased in frequency because hackers are more experienced and act way faster than in the past.

Examples of exploits

Among the most well-known web-based security vulnerabilities are SQL injection attacks, cross-site scripting, cross-site request forgery, and broken authentication code or security misconfigurations. In general, exploits can be classified into 2 main categories: known and unknown (or zero-day vulnerabilities).

What is a zero-day exploit?

The zero-day vulnerabilities are by far the most dangerous, as they occur when software contains a critical security vulnerability of which the vendor is unaware. The vulnerability only becomes known when a hacker is detected exploiting the vulnerability, hence the term zero-day exploit. Once such an exploit occurs, systems running the software are left vulnerable to an attack until the vendor releases a patch to correct the vulnerability and the patch is applied to the software.

What is a SQL injection exploit?

SQL injection is a code injection technique, used to attack data-driven applications, in which nefarious SQL statements are inserted into an entry field for execution (e.g. to dump the database contents to the attacker). SQL injection must exploit a security vulnerability in an application's software, for example, when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and unexpectedly executed.

Since an SQL Injection vulnerability could possibly affect any website or web application that makes use of an SQL-based database, the vulnerability is one of the oldest, most prevalent and most dangerous of web application vulnerabilities.

To run malicious SQL queries against a database server, an attacker must first find an input within the web application that is included inside an SQL query.

For an SQL Injection attack to take place, the vulnerable website needs to directly include user input within an SQL statement. An attacker can then insert a payload that will be included as part of the SQL query and run against the database server.

What is a directory traversal exploit?

The directory traversal/path traversal attack (also known as dot slash attack) is an HTTP exploit that allows an attacker to access restricted files, directories and commands that reside outside the web server's root directory. Directory traversal attacks are executed through web browsers. An attacker may manipulate a URL in such a way that the website will reveal the confined files on the web server.

This type of exploit owes its name to the fact that the attacker uses a special character "../"sequence to escape web document root, or alternate encodings of the "../" sequence to bypass security filters and access files or directories that reside outside the root directory.

The attackers can modify critical files such as programs or libraries, download password files, expose the source code of the web application, or execute powerful commands on the web server, which can lead to complete compromise of the web server.

How to prevent exploit attacks

The exploits we face today are more aggressive and spread throughout the system in a matter of minutes, compared to those in the early 90s, which were slower and passive because of the lack of internet connectivity. Now exploit kits are widely available for purchase on the Dark Web, as well as other malware, turning any script novice into a genuine schemer.

The problem with exploits is that they are part of a more complex attack, which makes them a nuisance. They never come alone and always infect your device with some form of malicious code.

Although security specialists and vendors work together to detect vulnerabilities as quickly as possible and issue patches to fix them, they can't always protect users from zero-day exploits. Worse, they can't protect users against their own negligence. You can take matters into your own hands and always back up your data, avoid weak passwords and constantly update all software. Never run vulnerable versions of the plugins, browsers, or media players. Remember that any minute you "waste" updating your operating system will save you hours of computer maintenance when disaster strikes.

Because exploits can spread through emails and compromised web pages, stay alert and be careful what you click on. Your computer's firewall and security software solution should be a good start for first-layer protection but remember that there is still a high risk of zero-day exploits.