The Ultimate Guide to Zero-Day Attacks & Exploits

March 1, 2021 0

The Ultimate Guide to Zero-Day Attacks & Exploits

Zero-Day Attacks Are Hard to Protect Against Because They Take Advantage of Previously Unknown Vulnerabilities – Learn All About Them and How to Minimize the Risks

It's finally here – the vacation you've been planning has arrived and it's time to hit the road. You lock your doors and head to the airport. But a few days later, while you're gone, a burglar decides to check out your neighborhood. They come to your house and, to their delight, find that the lock on the back door is broken. You didn't notice, but the bad guys did. They are thus able to steal your valuables before you have the chance to fix your door. Now, replace your house a piece of software and the broken lock with a vulnerability and you're looking at how zero-day exploits work in the cybersecurity world.

Zero-day attacks are extremely common and topped WatchGuard's list of the most popular types of network attacks in 2019. The report also found that zero-day exploits were responsible for half of all malware detections, a figure that increased 60% from the previous year. Likewise, the Ponemon Institute's Study on the State of Endpoint Security Risk determined that 80% of all successful data breaches in 2019 directly resulted from zero-day attacks. These stats are no surprise when you look at how much danger zero-day exploits pose and how effective they are at helping cybercriminals achieve their goals.

Most people think of zero-day exploits as tools created and used by black hat, criminal hackers. However, government agencies around the world are just as interested in obtaining them, most often for use in surveillance or their own cyberattacks. In fact, a thriving black market has emerged as a result of the fierce competition from both sides to be the first to know about zero-day vulnerabilities and the ways to exploit them.

So, what are zero-day attacks exactly? Why are they so dangerous? How are zero-day exploits used? And how can the damage be minimized?

Let's hash it out.

What Is a Zero-Day Attack/Exploit?

Zero-day exploits take advantage of vulnerabilities that unknown to the public-at-large at the time of their creation, including the developers of the vulnerable software and the makers of security tools like anti-virus and anti-malware programs. When a hacker uses a zero-day exploit to break into a system, the resulting hack is called a zero-day attack.

The "zero-day" portion of the name refers to the number of days that the good guys have to fix the problem, and zero-day exploits happen when the bad guys win the race to find any particular vulnerability. After locating a vulnerability, hackers then create an exploit that's able to make use of it during an attack. Since no one previously knew about the vulnerability, there will likely be minimal defenses in place to combat the exploit, making a zero day attack more likely to succeed. Once a vulnerability becomes publicly known, then the race is on for the software vendor to try and release a patch that fixes the problem before attackers can capitalize. (Side note: this is why security researchers and penetration testers are so important—their job is to find vulnerabilities before hackers do.) All too often, though, the vendor will find out only after an attack has occurred, and then they'll be trying to merely minimize the damage.

The most popular attack vectors for zero-day exploits are widely used programs such as web browsers and common file types (and the programs that handle them) such as Word, Excel, or PDF files. If you're going to spend all the time to create a zero-day exploit, you may as well target a program nearly everyone uses so you have lots of targets for your evil plan.

Zero-Day Vulnerabilities vs Exploits vs Attacks

It can be easy to confuse the three terms, and they are often incorrectly used interchangeably. Let's make sure their definitions are clear before moving forward:

Zero-Day vulnerabilities are security weaknesses in software, firmware, or hardware that are unknown to vendors. Accordingly, patches for them do not exist at the time of their release to the public. Usually, they only become known after an attack has occurred and has been investigated by researchers.

Zero-Day exploits take advantage of zero-day vulnerabilities. The exploit itself is the method and/or code that hackers use to carry out an attack against whatever possesses the vulnerability. They don't necessarily need to use their exploit immediately, either. They can create it and strategically wait for the best time or place to deploy it. It's still a zero-day exploit at that point since it hasn't become public knowledge yet.

Zero-day attacks are the final step, and they happen when the attackers finally decide to make use of the exploit. This is often done via zero-day malware that contains the exploit. At this point, it is only a matter of time before the zero-day vulnerability is discovered since the hacker's tools are now out in the open and able to be studied and reverse-engineered by researchers.

Why Are Zero-Day Attacks/Exploits so Dangerous?

As we mentioned earlier, zero-day exploits get their name from the fact that they've been known by vendors for exactly zero days. Usually, patches fix vulnerabilities that enable the creation of zero-day exploits, but unfortunately, they can't be created instantly. This gives attackers time to run rampant and carry out zero-day attacks on defenseless targets.

It can take days, weeks or even months for fixes to be released. Users could be forced to use compromised software that entire time, exposing their machines and personal data. Not to mention the fact that people are often slow to actually download and install new updates.

Zero-day exploits also drastically increase the everyday risks for the average user. Not only are software vendors unprepared for zero-day exploits, but so are the developers of your security software. Usually your anti-virus or anti-malware programs would catch any malicious malware encountered during routine internet activities. But with zero-day exploits, your machine has a much higher chance of getting infected while viewing websites, dealing with suspicious messages, or downloading third-party programs.

It used to be that a single zero-day exploit could be disastrous for an application. However, thanks to improved security mitigation in modern operating systems, it is now often necessary to chain together multiple zero-day exploits in order for an attack to be successful.

Typical Targets of Zero-Day Exploits

The kind of hackers that deal with zero-day exploits usually don't mess around. They go after bigger, high-value targets such as:

- Government agencies
- Large businesses and organizations
- Individuals with access to high-value information, such as confidential business data
- Software with large numbers of users such as operating systems or browsers
- Large groups of individual users for use in botnets
- Hardware including IoT devices and the associated firmware
- Political targets and/or national security threats

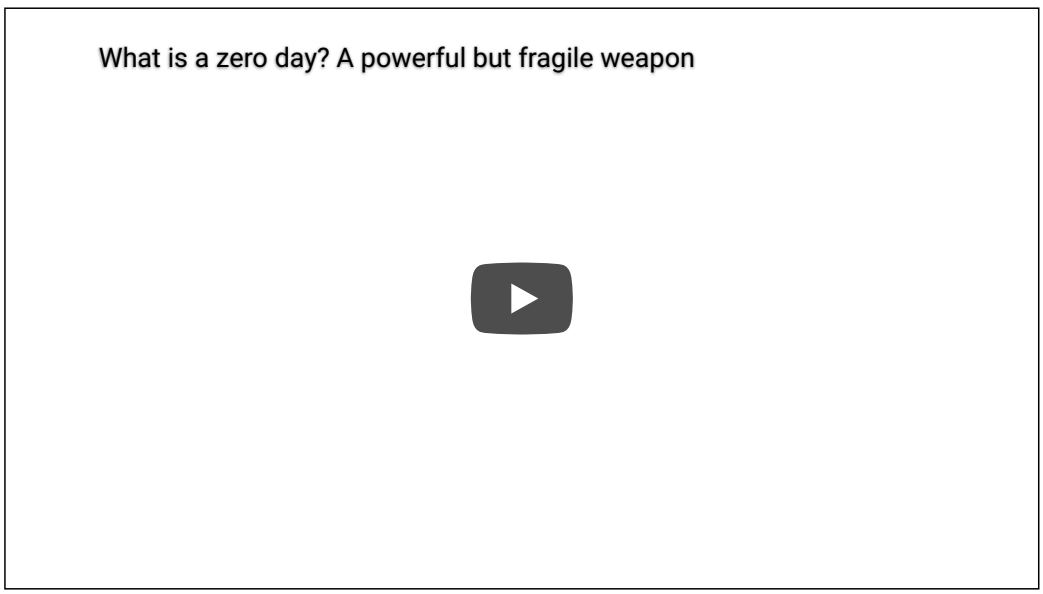
So, while attackers aren't usually going after specific individual users, that doesn't mean that the average person won't be affected by zero-day attacks. They often end up being collateral damage or used as tools in a grander scheme. Non-targeted attacks aim to hit as many users as possible, and in that case your personal data is just as valuable as the next person's, which means the dangers are still very significant.

The Process of Zero-Day Attacks

So, now that we've covered what zero-day attacks are and why they're so dangerous, let's take a look at the typical life of a zero-day exploit from beginning to end:

1. First, a vulnerability is created, unwittingly, by a software developer.
2. That software gets released, and eventually a hacker finds the vulnerability.
3. The hacker creates a zero-day exploit to take advantage of the vulnerability and deploys it via an attack while the vulnerability still exists in the code.
4. The vulnerability is discovered by the vendor (often because the zero-day attack was detected and reported by a security team that uses the vendor's software). At this time, however, they do not have a fix for it.
5. The vulnerability is disclosed, usually by the vendor and/or security researchers, and the public is warned of the dangers.
6. Usually, antivirus signatures are released next. In the event that zero-day malware is being used, security vendors can identify its signature and update their definitions to provide protection against it. There's still a chance that there are alternate ways to exploit the vulnerability, however, so users must still be careful.
7. The vendor releases a patch to close the vulnerability. This timeframe for this step can vary greatly depending on the complexity of the situation.
8. The patch is deployed to the userbase over time. The speed depends on how quickly the users perform the update.

The "zero-day attack" portion occurs between steps 2 and 5, but it's important to remember that systems can still be vulnerable during every step.



How Zero-Day Vulnerabilities are Found

Let's take a step back. How are the vulnerabilities used in zero-day attacks discovered in the first place? One method is via fuzzing, which is when a large amount of data is input to a program at varied intervals. The attacker then looks to see how the program responds. Overloading can lead to crashes or strange behavior, which can in turn expose bugs. Finally, the attacker writes their code to reproduce the behavior that led to the bug, which is how the exploit itself gets created.

We've all heard the phrase "those that don't learn from the past are doomed to repeat it." Well, attackers sometimes will try and learn from the past so they CAN repeat it. They do this by studying previous exploits and try to adapt the same methodology to newer software. Hackers will also analyze new patches and updates to try and gain insight into the inner workings of the program in the hope of finding vulnerabilities.

It's not just the bad guys that use these methods. Developers themselves will do similar during the dev process to try and find vulnerabilities before they get released to the rest of the world.

Examples of Zero-Day Attacks

Here are a few notable examples of zero-day attacks, along with how they worked and the impact they had on their victims:

- **Stuxnet** – This malicious computer worm was discovered in 2010 and affected PLCs used in Iran's uranium enrichment plants. Specifically, the Siemens Step7 software was exploited. The program ran on Windows and was used to program the PLCs. The exploit caused unexpected commands to be executed by the machinery and disrupted the centrifuges used to process nuclear material. The ultimate goal was to damage the nation's nuclear program.
- **RSA** – In 2011, the network of security company RSA was infiltrated by hackers thanks to a vulnerability in Adobe Flash Player. Compromised Excel spreadsheet files containing malicious Flash files were distributed to RSA employees. Upon opening, the Poison Ivy remote admin tool took control of the user's machine, providing access to RSA's network. The attackers then proceeded to steal sensitive data related to the company's SecurID two-factor authentication products.
- **Zoom** – In 2020, a vulnerability was found in the popular video-conferencing platform. It allowed an attacker to remotely access a user's PC if they were running an older version of Windows. If the victim was an administrator, then the hacker could access all files and essentially take over the machine completely.

As you can see, the targets of these attacks fall in-line with what we discussed earlier. They were either a national/political target (Stuxnet), a company with valuable intellectual data (RSA), or a piece of software that has an extremely large userbase (Zoom).

The Market for Zero-Day Exploits

The kind of exploits we just talked about are always in high demand and can command astronomical prices on the open market. Want to make a few hundred thousand, or even millions of dollars? Just find a zero-day exploit for the right piece of software and you'll be all set.

Just kidding, we certainly aren't advocating the sale of exploits on the black market. Ideally, those that discover vulnerabilities would immediately report them to the software vendor so they could be patched as soon as possible. Some companies even have bug-bounty programs where they will pay for knowledge regarding vulnerabilities in their products.

Overall, the market for zero-day exploits can be broken down into three different segments:

- **The black market** – This is where criminal activity occurs. Exploits are used for nefarious means such as the theft of confidential data or gaining unauthorized access to networks.
- **The white market** – This is where "the good guys" operate. Researchers willingly report vulnerabilities to vendors and may be financially rewarded for their work. Bug-bounty programs fall under this category.
- **The grey market** – This one is specific to the military. Exploits are sold for use in national security-related activities, including surveillance, technological warfare, and espionage.

Detecting Zero-Day Vulnerabilities

There are a few different things developers look for that can tip them off to the existence of a zero-day vulnerability:

- **Abnormal behavior of software** – Vendors routinely examine how applications respond to previous exploits and will try to apply that knowledge to new products. Attacks often result in the emergence of patterns that can be used to detect future exploits.
- **Trends** – First off, attacks are more likely to happen immediately after a security update is released. This is true with Microsoft-related vulnerabilities in particular because they employ a monthly update cycle. By attacking the day after an update, the bad guys will have nearly a month to use their exploit before it gets patched. The statistics of previous attacks can also be leveraged – for instance if the volume of data being transmitted during an attack had a unique spike, that intel can be useful in the future.
- **Signatures of previous vulnerabilities** – These signatures are like fingerprints, and it's possible that certain characteristics will appear again. By scanning for them, vulnerabilities can be located and removed.

None of these methods are comprehensive or fool-proof, so they are usually used together.

Issues with the Patching of Zero-Day Exploits

We've touched on the patching process and how that theoretically solves the issue that zero-day exploits create. Realistically, though, a significant number of devices will never get patched. Its especially bad with IoT devices. Just think about it – you have a large quantity of widgets that get shipped out of the factory containing a vulnerability. With some devices, it can literally be impossible to update them. Other times, it might be impractical to assume that users will perform an update.

Take a smart coffee maker for instance – realistically, most people aren't going to go to the trouble of performing an update for something that they think of as inherently safe (even though attackers could potentially use its wi-fi capabilities to access other machines on the same network). So, even though the zero-day exploit may be theoretically fixed (and no longer a true "zero-day"), it can still be useful to attackers in the long term.

Protecting Yourself from Zero-Day Attacks

Because of their nature, it's impossible to 100% protect yourself from zero-day exploits. How can you protect yourself from something you don't even know exists? The good news is there are several best practices that you can follow to minimize the risk of a zero-day attack, at least:

- Practice secure software lifecycle development to keep your code safe and secure
- Institute vulnerability management and patching programs so that software gets updated right away
- Deploy layered security controls and limit access to the smallest number of users possible
- Have comprehensive disaster recovery and back-up plans (using something like CodeGuard that allows for the instant restoration of your site)
- Keep an eye out for announcements regarding zero-day exploits and install patches as soon as they're available
- Perform vulnerability scanning, which can sometimes detect zero-day vulnerabilities
- Use penetration testing to rigorously test your code/software for weaknesses and vulnerabilities
- Implement firewalls, intrusion detection, and other systems that allow you to immediately block and/or respond to a wide variety of attack types
- Perform input validation and sanitization to close the attack vector that comes with input fields on sites and applications
- Create a bounty program that rewards researchers for finding vulnerabilities in your software

Zero Risk of Zero-Day Attacks (or as Close as Possible, at Least)

Zero-Day attacks occur without warning, which make them difficult to protect against. The fact they usually go after high-profile targets makes them all the more threatening. By following general security best practices and having a back-up plan in place, though, you can help reduce the potential damage that could occur. Always apply patches as soon as possible too, which will minimize the window of danger for any given vulnerability. By following these suggestions, your risk of exposure to zero-day attacks will be as close to zero as possible.

Be the first to comment

Leave a Reply

Your email address will not be published. We will only use your email address to respond to your comment and/or notify you of responses. Required fields are marked *

Comment

Name *

Email *

Website

Notify me when someone replies to my comments

Captcha *

6 − = one

POST COMMENT

Author

Mark Vojtko

After starting his career as an engineer, Mark pivoted to tech marketing, which combines his love of technology and analytical thinking with a generous dose of creativity. In addition to contributing to Hashed Out, Mark is The SSL Store's Product Marketing Manager.

Recent Posts

Smile! Thanks to Verkada Breach, You Could Be on Candid Camera
March 23, 2021

The Ultimate Guide to Stored XSS Attacks
March 19, 2021

Password Security: What Your Organization Needs to Know
March 15, 2021

New Browser Hack Lets Sites Track You Via Favicons
March 12, 2021

Post-Covid WFH Shadow IT: A Concern or Opportunity?
March 4, 2021

Follow Us

Free Ebooks

Email Security Best Practices – 2019 Edition

Download Now

Certificate Management Best Practices Checklist

Download Now