Analysis of growing Zero Day Vulnerability

[1]V.Ashishya [2]Mr.N.Naveen Kumar

[1]M.Tech Student, School of Information Technology JNTUH, Kukatpally, Medchal-Malkajigiri, Hyderabad.

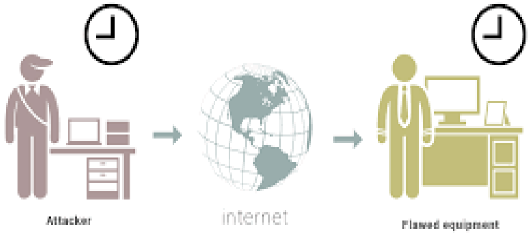
[2]Assistant Professor, School of Information Technology JNTUH, Kukatpally, Medchal-Malkajigiri, Hyderabad.

Abstract: Zero-day vulnerability is vulnerability in a system or device that has been disclosed but is not yet patched. An exploit that attacks zero-day vulnerability is called a zero-day exploit. A zero-day exploit can occur in one of several ways. Usually, it is carried out in five steps, which can be done just minutes after a security hole becomes available. The features, detection and preventive measures are discussed in this paper. Preventing zero-day attacks is the only possible way as exploiting is not possible. Zero day attacks are increasing day by day. Few Cybersecurity Ventures, predicted that by 2021, there will be one new exploit every day. In 2015, there was about one per week. The most recent zero day attack on Microsoft which was revealed through Twitter is also discussed the paper.

1. INTRODUCTION

Zero-day vulnerability is vulnerability in a system or device that has been disclosed but is not yet patched. An exploit that attacks zero-day vulnerability is called a zero-day exploit. A zero-day exploit is different from zero-day vulnerability. Zero-day exploits do not have to be existing vulnerabilities: they could be a brand new malware of ransomware program. A zero-day exploit is a brand new kind of attack in progress that requires immediate remediation. When a zero-day vulnerability isn’t discovered and patched before the attackers find the flaw, however, it becomes a zero-day exploit as well. Zero-day exploits are difficult to detect and defend against: they are unknown until it’s too late, and their nature is under-researched. Signature-based security solutions can’t detect a zero-day exploit, and there are no software vulnerability patches immediately available. You need to react to zero-day exploits quickly to prevent widespread damage to the network or data theft. Thus, a zero day attack is the term used to describe the threat of unknown security vulnerability in a computer software or application for which either the patch has not been released or the application developers were unaware of or did not have sufficient time to address.

A zero-day exploit can occur in one of several ways. Most often, the attack is enabled by a hole in some programming code that the hacker discovers before the programme has time to react. In cases like these, the hacker will be hours, if not days, ahead of the programmers, who likely won't even realize there is a breach until thousands of users have already been affected. In some cases, the malware creator will spread the infection via links in widely distributed emails.



As soon as a user clicks on the link, his or her computer is infected with a code that allows the hacker to view and retrieve the unsuspecting person's data. A zero-day exploit is usually carried out in five steps, which can be done just minutes after a security hole becomes available:

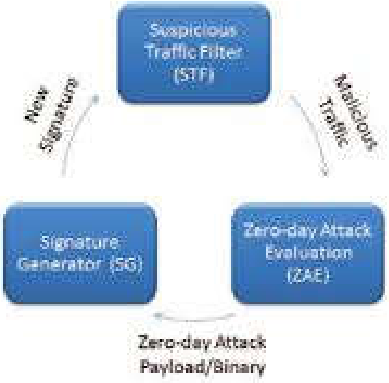
1. Scan for Vulnerabilities: Hackers scan the codes of new software programs in search of vulnerabilities. In certain scenarios, the exploits are exchanged between hackers.

2. Spot a Security Hole: Once a weakness has been spotted within a programming code, the attacker knows where to hit.

3. Create an Exploitation Code: Now that the hacker knows the nature of the weakness, the game is on to develop a malware to take advantage of the situation.

4. Infiltrate the System: Before the software developer discovers the hole or has time to react, the hacker must now slip past detection to infect the system.

5. Launch the Exploit: With the malware code now ready, the hacker plants the virus on the Internet.


Suspicious Traffic Filter (STF)
Malicious Traffic
Zero-day Attack Evaluation (ZAE)
Zero-day Attack Payload/Binary
Signature Generator (SG)
New Signature

The main reason zero-day attacks occur is twofold. For starters, the fact that a security hole exists in the first place gives the hacker a unique opportunity. Second, the hacker will usually have a decent window of time to exploit the situation, since they usually are the one to first spot the hole. Even in cases where software developers discover the problem, the hacker will still have time to exploit the weakness before a patch is created.

Another problem is the time it can take to develop a patch that successfully fills a security hole. Programmers might know about a breach for weeks before they can develop a foolproof code and put an end to the problem. By that time, an untold number of computers could be infected.

2. RELATEDWORK

Recent Microsoft Windows zero-day vulnerability disclosed through Twitter bug being actively exploited in the wild, as part of its Patch Tuesday security bulletin. The vulnerability is an elevation-of-privilege flaw, rated important, affecting the Windows Win32k component. The zero-day (CVE-2018-8453), found by Kaspersky Lab, could allow an adversary to run arbitrary code in kernel mode on targeted systems. “An attacker could then install programs; view, change or delete data; or create new accounts with full user rights,” Microsoft wrote in its patch update. Windows 7, 8.1, 10, and Server 2008, 2012, 2016, and 2019 are affected. Middle East-based APT FruityArmor, which has a history of targeting Windows zero-day, is believed to be actively exploiting the flaw, according to Kaspersky Lab. In 2016, Kaspersky Lab researchers reported that the group carried out a number of targeted attacks exploiting zero-days to escape browser-based sandboxes and execute malicious code in the wild. In that case, the adversaries targeted CVE-2016-3393, tied to Windows graphics device interface. The zero-day patch was one of 49 fixes issued Tuesday; 12 were listed as critical.

Microsoft also patched an eight-year-old remote code-execution vulnerability, first identified in 2010 and rated critical. The bug (CVE-2010-3190) is tied to a nagging issue with Microsoft Foundation Class Library, a resource used by developers to manage how DLL files are loaded and handled by an application. The bug has been patched multiple times over the years: in 2010, 2011 and 2016 with the most recent update available Tuesday. Microsoft said the problem is once again an issue as it relates to installations of Exchange Server 2016. “An attacker who successfully exploited this vulnerability could take complete control of an affected system. An attacker could then install programs; view, change or delete data; or create new accounts with full user rights,” Microsoft wrote. The software giant added, “Exchange Server was not identified as an in-scope product when CVE-2010-3190 was originally published…The update addresses this vulnerability by correcting how applications built using MFC load DLL files.” Microsoft continued the trend from last month where they introduced both a monthly roll-up and a security-only release for Server 2008,” he said. Microsoft’s ubiquitous Office Suite bundle also received a number of updates including those for Excel, Outlook, PowerPoint and Word.

3. FRAMEWORK

The salient features of the zero day or day zero attacks are:

• Zero day attacks usually occur between the time the vulnerability is first found and exploited and the time the application developers releases the necessary solution to counter the exploitation.

This timeline is usually termed as the vulnerability window.

• Zero day attacks are capable of devastating a network by exploiting the vulnerabilities of the applications involved.

• They are not always viruses and can assume other malware forms such as Trojan horses or worms.

• For home computer users, the zero day attack is extremely difficult to diagnose as the nature of attack is through a trusted entity.

• Update of latest anti-malware software are often recommended, though it can only provide a minimum security against a zero day attack.

Techniques for detecting zero-day exploits **Statistics-based detection** techniques rely on data about previously detected exploits inside a particular system. Statistics-based detection solutions often employ machine learning to aggregate statistical data on past exploits and determine a baseline for safe system behavior. The main advantage of such solutions is that the more data they have, the more accurate they become. As a statistics-based solution runs within a system, it gathers more information about new zero-day exploits, thus expanding its dataset and producing a more sophisticated profile for a potential new exploit. However, depending on the baseline chosen, such a solution may also produce a high number of false positives and false negatives. It may be hard for developers to find the right balance with the baseline since on the one hand false negatives need to be avoided in order not to miss a zero-day attack, while on the other hand the number of false positives needs to be minimized to avoid impacting the daily operations of the company. Overall, the effectiveness of statistics-based techniques for zero-day exploit detection is limited. They also have limited capabilities for detecting malware with heavily encrypted and obfuscated code. However, statistics-based techniques can work well as part of a hybrid solution, which we’ll cover further down. **Signature-based detection** techniques are usually employed for malware detection by [legacy] antivirus software. As the name implies, the technique relies on existing databases of malware signatures, which are used as a reference when scanning a system for viruses.

Although signature databases are usually updated very quickly, they cannot be used to detect new zero-day attacks since, by definition, zero-day exploits don't have a known signature. Thus, the only way to use signature-based detection for defense against zero-day attacks is to use machine learning and similar algorithms to generate signatures in real time that might match a currently unknown malware and thus be able to detect it.

There are three types of signatures that can be generated this way:

- Content-based – a signature based on components typically present in most exploits (such as certain parts of code)

- Semantic-based – a signature based on typical actions taken by malware

- Vulnerability-based – a signature based on establishing the conditions for a vulnerability and how easily achievable they are; vulnerability-based signatures usually use data on known vulnerabilities to establish a baseline, and therefore the accuracy of the baseline is determined by the size of the data pool.

Overall, the ability to quickly generate accurate signatures that correspond to real-life malware is what determines whether a signature-based approach actually works for detecting zero-day exploits.

Behavior-based detection techniques look for characteristics of malware based on the way it interacts with the target system. This means that a solution using a behavior-based technique doesn't examine the code of incoming files, but instead looks at the interactions they have with existing software and tries to predict whether this is the result of any malicious actions. Machine learning is often used to establish baseline behavior based on data of past and current interactions within the system. As with statistics-based detection techniques, the more data is available, the more reliable the detection becomes. A behavior-based detection system that works on a single target system for a long time may prove very effective in predicting results of current processes and actually detecting malicious software.

Hybrid detection techniques are aimed at taking advantage of the different strengths of the three techniques mentioned above while at the same time avoiding their weaknesses. Hybrid detection solutions usually combine two or three techniques in a way that allows them to produce more accurate results. For example, a statistics-based algorithm can be used to reinforce a behavior-based baseline for normal behavior and to speed up the learning process, while a signature-based approach can be used to filter false positives, increasing the accuracy of detection. Due to their effectiveness, popularity of hybrid-based solution is constantly rising.

4. EXPERIMENTS

Measures to Prevent Zero-Day Attacks

Zero-day vulnerability will open your system to the possibility of an instant attack that could have disastrous results and grave financial consequences. Therefore, it’s crucial to be alert to this possibility and act if and when vulnerability does appear. Some steps you can take include:

• Employing the Most Advanced Security Software: Basic security software is simply not enough in today’s online climate, where hackers employ the most advanced means of system hacking. A software that only protects against known threats is no match for the hacker who develops new ways to attack.

• Keeping Security Software Up-to-Date: As new methods of hacking become known, security software is updated to prevent such hacks. Only with regular, timely software updates can you effectively protect your network from a zero-day exploit.

• Updating Your Browsers: Web browsers are among the most common targets of hackers. If your browser is out-of-date, it could be vulnerable to malware that did not exist when you first updated to that version of the browser. Even though today’s browsers — such as Firefox, Chrome and Opera — usually update automatically, you should still check periodically to ensure all the computers in your network are equipped with the latest version of each browser.

• Implementing Security Protocols: For a network to be fully ready to act on zero-day vulnerability, all company personnel must be trained on the best practices for security. Develop and implement a sequence of security measures and teach your workforce about when and how to enact these measures.

On rare occasions, even the most diligent organization can have its security compromised by a zero-day exploit. In the event of this happening to your company, make sure all your security measures are ready to go at a moment’s notice. Even though it can take hours or days to develop a security patch, the spread of a worm or virus can immediately be halted if connections are limited to their barest essentials. The moment an infection becomes known, shut off all network connections that are not vital to the function of your business. This will block the spread of the virus and give your company time to assess the problem and develop a solution.

5. CONCLUSION

Overall, zero-day attack prevention and detection are extremely difficult problems, but there’s no denying the high demand for solutions in these areas. The practice of issuing monetary rewards for reported exploits has become nearly universal, and the amount of money offered is constantly going up. Thus, if you decide to invest in a Cybersecurity solution, zero-day exploit detection is one of the great areas to focus on.

REFERENCES

[1] L. Ablon, A. Bogart, “Zero Days, Thousands of Nights: The Life and Times of Zero Day Vulnerabilities,” RAND Corporation (2017).

[2] A. Greenberg, “Software Has a Serious Supply-Chain Security Problem,” WIRED Sept 18, 2017.

[3] Kaspersky Lab, “ShadowPad: How Attackers hide Backdoor in Software used by Hundreds of Large Companies around the World,” August 15, 2017.

[4] M. Gorelik, “Morphisec Discovers CCleaner Backdoor Saving Millions of Avast Users,” Morphisec Cyber Security Blog, Sept 18, 2017. http://blog.morphisec.com/morphisec-discovers-ccleaner-backdoor (retrieved Sept 21, 2017).

[5] Talos, “CCleanup: A Vast Number of Machines at Risk,” Talos Intelligence Blog, Sept 18, 2017. http://blog.talosintelligence.com/2017/09/avast-distributes-malware.html (retrieved Sept 21, 2017).

[6] S. Checkoway, J. Maskiewicz, C. Garman, J. Fried, S. Cohney, M. Green, N. Heninger, R.-P. Weinmann, E. Rescorla, H. Shacham, “A Systematic Analysis of the Juniper Dual EC Incident,” Proceedings of the ACM Conference on Computer and Communications Security, pg. 468-479 (2016).

[7] D. Goodin, “Devs unknowingly use “malicious” modules snuck into official Python repository,” ARS Technica Sept 16, 2017. https://arstechnica.com/information-technology/2017/09/devs-unknowingly-use-malicious-modules-put-into-official-pythonrepository/ (retrieved Sept 21, 2017).

[8] B. Delamore, R.K.L. Ko, “A Global, Empirical Analysis of the Shellshock Vulnerability in Web Applications,” IEEE Trustcom/BigDataSE/ISPA (2015).

[9] M. Carvalho, J. DeMott, R. Ford, “Heartbleed 101,” IEEE Security and Privacy, vol. 12, iss. 4, pg 63-67 (2014).

[10] D. Goodin, “Extremely severe bug leaves dizzying number of software and devices vulnerable,” ARS Technica Feb 16, 2016 http://arstechnica.com/security/2016/02/extremely-severe-bug-leaves-dizzying-number-of-apps-and-devices-vulnerable/ (retrieved Oct 14, 2016).