DOI: 10.36910/6775-2524-0560-2020-39-25
УДК: 004.05(075.8)
Марценюк Василь Петрович, д.т.н., професор,
https://orcid.org/0000-0001-5622-1038
Університет Бєльсько-Бяли, Польща[1)]
Дідманідзе Ібраім Шотаевич, д.фіз.-мат.н., професор,
https://orcid.org/0000-0001-6695-4980
Батумський державний університет імені Шота Руставелі, Грузія[2)]
Сверстюк Андрій Степанович, к.т.н., доцент,
https://orcid.org/0000-0001-8644-0776
Тернопільський національнийний медичний університет імені І.Я. Горбачевського, Україна[3)]
Андрущак Ігор Євгенович, д.т.н., професор,
https://orcid.org/0000-0002-8751-4420
Рудь Катерина Іванівна, аспірант.
https://orcid.org/0000-0002-1148-6080
Луцький національний технічний університет, Україна[4)]

AUTOMATED METHOD OF BUILDING EXPLOITES IN ANALYSIS SOFTWARE TESTING.

Марценюк В.П., Дідманідзе І.Ш., Сверстюк А.С., Андрущак І.Є., Рудь К.І. Автоматизований метод побудови експлойтів в аналізі тестування програмного забезпечення. У статті розглядається проблема методу автоматизованої побудови експлойтів для уразливості переповнення буфера на стеку і його застосування до задачі оцінки критичності помилок у програмному забезпеченні і побудові захисту програмних засобів. Проаналізовано сучасні практичні і теоретичні методи вторгнення в різні операційні системи.
Ключові слова: безпека цільової системи, експлойт, інформаційна безпека, переповнення системного буфера, несанкціонований доступ, вразливість в програмному забезпеченні, класифікація помилок.

Марценюк В.П., Дидманидзе И.Ш., Сверстюк А.С., Андрущак И.Е., Рудь К.И. Автоматизированный метод построения эксплойтов в анализе тестирование программного обеспечения. В статье рассматривается проблема метода автоматизированного построения эксплойтов для уязвимости переполнения буфера на стеке и его применение к задаче оценки критичности ошибок в программном обеспечении и построении защиты программных средств. Проанализированы современные практические и теоретические методы вторжения в различные операционные системы.
Ключевые слова: безопасность целевой системы, эксплоит, информационная безопасность, переполнение системного буфера, несанкционированный доступ, уязвимость в програмном обеспечении, классификация ошибок.

Martsenyuk V.P., Didmanidze I.Sh., Sverstiuk A.S., Andrushchak I.Ye., Rud K.I. Automated method of building exploites in analysis software testing. The article discusses the problem of the method of automated construction of exploits for the buffer overflow vulnerability on the stack and its application to the task of evaluating the criticality of errors in software and building software protection. The modern practical and theoretical methods of invading various operating systems are analyzed.
Keywords: target system security, exploit, information security, system buffer overflow, unauthorized access, software vulnerability, error classification.

Formulation of the problem. In modern distributed information systems, various types and nature of threats prevail associated with unauthorized access and data leakage. There are threats that are aimed at harming the personal data of the user through their damage or copying for their personal benefit for the purposes of using directly against the user himself. The standard definition describes an "exploit" as a program or code that exploits security flaws in a specific application to infect a device. Users may mistakenly consider this to be separate malware. However, in fact, this is a piece of program code that allows you to penetrate the computer system and affect its operation. Using a specific vulnerability, this tool provides attackers with the necessary permissions to launch malicious components and infect the system.

Security solution developers often mention exploits in their publications as one of the most serious data and system security issues, although it is not always clear what the difference is between exploits and malware in general.

At the development stage, all programs and networks incorporate protection mechanisms against hackers, such as locks, preventing unauthorized attacks from outside. The vulnerability is similar to an open window, to get through which is not difficult for an attacker. In the case of a computer or network, cybercriminals can install malicious software by exploiting a vulnerability in order to gain control or infect the system for their own personal gain with corresponding consequences. Most often, all this happens without the knowledge of the user.

Analysis of research. Attackers are constantly improving their tools and finding new ways to infect a large number of devices. One of the common methods for malware to infect victims' computers has been the use of exploits that ensure the rapid spread of threats. They also allow you to access programs and

subsequently infect the user's device through vulnerability in the security system. In recent years, the most active have been threats that exploit vulnerabilities in Java products, in Adobe software, and in the Windows operating system.

Recently, exploits have been used in many well-known cyber attacks. An example is the massive attack of the WannaCryptor virus (or WannaCry), which has become the largest digital threat in the world in recent years. It is worth noting that during this attack, the EternalBlue exploit was used, which was allegedly stolen by a group of cybercriminals at the National Security Agency (NSA). EternalBlue was aimed at the vulnerability of the implementation of the SMB protocol in an irrelevant version of Microsoft. In addition, EternalBlue was also a tool during the famous Diskcoder.C attack (Petya, NotPetya and ExPetya).

Browsers along with Flash, Java, and Microsoft Office are among the most attacked software categories. Because of their ubiquity, both security experts and hackers are actively exploring them, and browser developers are forced to regularly issue patches to fix vulnerabilities. It is best to install these patches at once, but, unfortunately, this does not always happen - after all, you will have to close all the tabs.

A special problem, of course, is the exploits of unknown vulnerabilities discovered and used by criminals - the so-called zero-day vulnerabilities. It may take a long time before manufacturers find out about the problem and fix it.

Presentation of the main material and the justification of the results. Depending on the method of gaining access to the vulnerable software, exploits are divided into remote and local. Remote exploit works through the network and exploits a security vulnerability without any prior access to the vulnerable system; a local exploit runs directly on the vulnerable system, requiring prior access to it. Usually used to gain administrator / superuser privileges.

Exploits are caused by errors in the software development process, which result in vulnerabilities in the program protection system that are successfully used by cybercriminals to gain unlimited access to the program itself, and through it further to the entire computer. Exploits are classified according to the type of vulnerability used by the hacker: zero day, DoS, spoofing, or XXS. Of course, program developers will soon release security updates in order to eliminate the defects found, but until this moment the program is still vulnerable to attackers.

Exploits are a subtype of malware. The term is associated with the English verb "to exploit", meaning "exploit, use in their own interests." However, an exploit is not necessarily a separate application (executable file): it can take the form of a small piece of malicious code or a set of commands executed in a certain order. Using the vulnerability in any system or application program, the exploit performs an unauthorized action on the victim's device. As a rule, it allows you to increase privileges on the target system or execute arbitrary code.

The objectives of the exploit are varied: downloading and installing malware, increasing access rights, stopping the system, and disclosing confidential data. It should be noted that the exploits themselves do not perform destructive actions directly: their task is to take advantage of the vulnerability to ensure that the code embedded in them is launched. All subsequent operations are carried out by the malicious load, and what the attacker will achieve depends on its content [1].

Since exploits exploit flaws in program security mechanisms, the average user has virtually no chance of determining their presence. That is why it is extremely important to keep installed programs up-to-date, especially in a timely manner, to install security updates issued by program developers. If the software developer releases a security update to eliminate a known vulnerability in his software, but the user does not install it, then, unfortunately, the program will not receive the latest virus definitions.

Exploit - a computer program, a piece of program code or a sequence of commands that exploit vulnerabilities in software and are used to carry out an attack on a computer system. The purpose of the attack can be both the seizure of control over the system (privilege escalation) and the disruption of its functioning (DoS attack). Exploits are actually designed to perform third-party actions on a vulnerable system and can be distributed in the form of source code, executable files, or a verbal description of the vulnerability exploitation.

After the vulnerability has been closed by the manufacturer, the chance of a successful exploit is rapidly decreasing. Therefore, the so-called 0day exploits that use recently appeared vulnerabilities that have not yet become generally known are especially popular among hackers [2].

The next part is quite technical, so feel free to skip, unless you are really interested in how it works. Keep in mind that cybercriminals often prefer exploits to other methods of infection, because, unlike social engineering, in which everything is done at random, exploiting vulnerabilities invariably gives the desired result.

There are two ways to feed exploits to users. Firstly, when they visit a site containing malicious exploit code. Secondly, when a user opens a file with a harmless-looking file with hidden malicious code. As you might guess, in the second case, spam or phishing emails are usually used to deliver an exploit.

Exploits take advantage of software vulnerabilities. Vulnerability is a hole in your software that can be used by malware to migrate to your device. Malicious programs exploit these vulnerabilities to prevent security threats from infecting your computer.

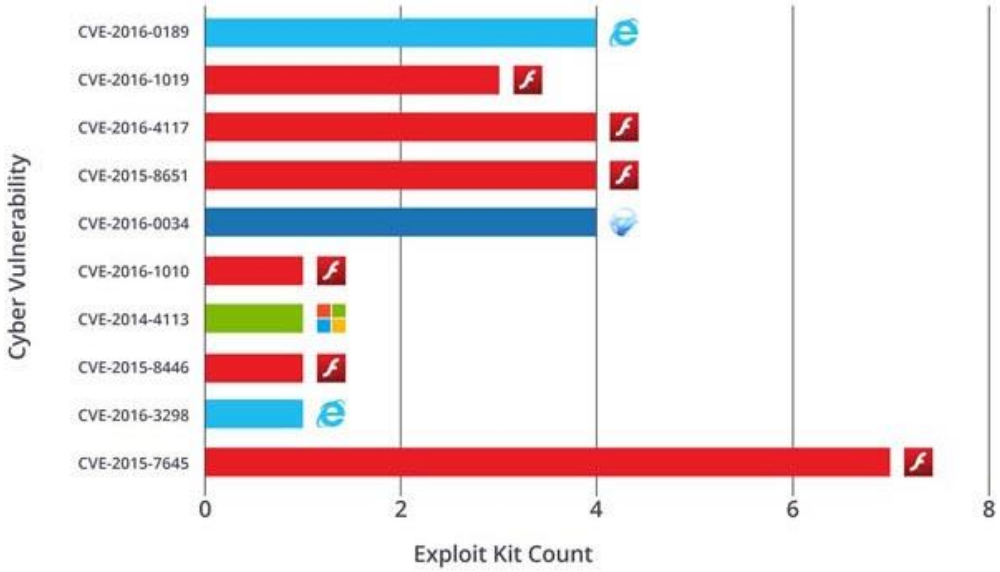
Most often, the use of exploits is the first part of a wider attack. Hackers scan outdated systems that contain important vulnerabilities that they then use to deploy targeted malware. Frequently used exploits include what is called shellcode. This is small useful malware data that is used to download additional malware from networks controlled by cybercriminals. This allows hackers to infect devices and infiltrate organizations.

Exploit packages are more complete tools that contain a set of exploits. These kits are designed to scan devices for vulnerabilities of various types of software and, if detected, deploy additional malware to further infect the device. Kits can use exploits designed for various software products, including Adobe Flash Player, Adobe Reader, Internet Explorer, Oracle Java and Sun Java.

The most common way to distribute exploits and exploit kits is through web pages, but exploits can also be emailed. Some websites are unknown and inadvertently contain malicious code and exploits in their advertising materials. The infographic below shows how a set of exploits can try to use the device when visiting a compromised web page.

Exploits are designed to attack specific versions of software containing vulnerabilities. Thus, if the user has the right version of the software when opening a malicious object, or if the website uses this software to work, the exploit is launched. Once it is accessed through a vulnerability, the exploit downloads additional malware from the server of criminals that carry out subversive activities, such as identity theft, using a computer as a botnet element to send spam, or perform DDoS attacks, and so on. Exploits pose a threat even to cautious and conscientious users who regularly update their software. The reason lies in the temporary gap between the discovery of the vulnerability and the release of the patch to fix it [3].

In this interval, exploits are free to operate and threaten the security of almost all Internet users in the absence of automatic means to prevent exploit attacks installed in the system. Again, let's not forget about the open tabs syndrome - timely updating of programs often requires some sacrifices from the user, which not everyone is ready to go right at the time the patch is released (Pic 1).



Pic 1. Vulnerability adoption by Exploit Kit

Depending on the method of gaining access to the vulnerable software, exploits are divided into remote (local) and local (local):

- remote exploit works through the network and exploits a security vulnerability without any prior access to the vulnerable system;
- a local exploit runs directly on the vulnerable system, requiring prior access to it. Usually used to gain root privileges by an attacker.

An exploit attack can target various components of a computing system - server applications, client applications, or operating system modules. To exploit the server vulnerability, it's enough to form an exploit and send a request to the server containing malicious code. Using a client's vulnerability is a bit more complicated - you need to convince the user of the need to connect to a fake server (click the link if the vulnerable client is a browser) [4].

Exploits are actually designed to perform third-party actions on a vulnerable system and can be divided among themselves as follows:

- exploits for operating systems
- exploits for application software (music players, office suites)
- exploits for browsers (Internet Explorer, Mozilla Firefox, Opera)
- exploits for online products (IPB, WordPress, VBulletin, phpBB)
- exploits for internet sites (facebook.com, hi5.com, livejournal.com)
- other exploits

The exploit can be distributed in the form of source codes, executable modules, or a verbal description of the exploitation of the vulnerability. It can be written in any programming language (most commonly used: C / C ++, Perl, Python, PHP, HTML + JavaScript)

Exploits can also be classified by the type of vulnerability they use, such as: buffer overflows, SQL code injection, crossite scripting, crossite request forgery

Information obtained as a result of vulnerability detection can be used both to write an exploit and to eliminate the vulnerability. Therefore, both parties are equally interested in it - both the cracker and the hacked software manufacturer. The nature of the distribution of this information determines the time it takes the developer to release the patch. Bundles of exploits are a package of exploits for several programs (versions) at once and / or for different vulnerabilities in them. In the latest versions of bundles, an exploit is selected specifically for a specific user program. In most cases, exploit kits are used for attacks that use browser vulnerabilities or their add-ons (common targets, for example, are Java, Flash and PDF). There are also sets of local exploits for elevating privileges in an attacked system. In fact, such sets are also bundles, but in a hacker environment they are not considered and are not called.

Exploits are created by highly qualified cybercriminals who sell them on the black market to other attackers. As elements of cyber weapons, they are developed and used by special services. Also, exploits may be the result of the work of information security specialists who want to show how the vulnerability they discovered can be exploited. In this case, software vendors are notified of vulnerabilities before the exploit is published in the public domain [5].

Exploits are often packaged together - so as to test the target system for a wide range of vulnerabilities. Once one or more is discovered, the appropriate exploits come into play. Exploit kits also make extensive use of special code obfuscation techniques (experts call it the clever word "obfuscation") to avoid finding and replacing Internet addresses to prevent researchers from calculating them. Exploits can be applied to any elements of a computer system. The target of an attack can be operating system modules, application programs, and even hardware components. For a successful attack, it is necessary to force the victim to follow the link, download and open the file so that the exploit is able to exploit the desired vulnerability.

We list a few of the most famous exploit kits, or, as they are also called, exploit kits [6-7].

Angler is one of the most sophisticated kits on the black market. This set of exploits, by its appearance, changed the rules of the game after it began to detect antiviruses and virtual machines (often used by security experts as decoys) and use encrypted files to make research difficult. This is one of those exploit kits that most quickly includes the newly discovered zero-day vulnerabilities in its arsenal, and its malicious programs work in memory, without writing to the victims' hard drives.

Nuclear Pack - Affects victims with Java and Adobe PDF exploits, and also spoils Caphaw, the notorious banking trojan.

Neutrino is a suite from Russian-language developers containing several Java exploits. Neutrino became famous last year due to the fact that the owner put it up for sale at a very modest price - $ 34 thousand. Most likely, this was done after the arrest of a certain Paunch, the creator of the next set, which we want to talk about.

The Blackhole Kit is the most common web threat in 2012, targeting vulnerabilities in older versions of Firefox, Chrome, Internet Explorer, and Safari, as well as many popular plugins such as Adobe Flash, Adobe Acrobat, and Java. After the victim has been lured or redirected to the landing page, the confusing JavaScript detects the contents of the victim's machine and loads the exploits for which this computer is vulnerable [8].

Despite the efforts of specialists, there are vulnerabilities in almost all programs, which means that there is always a loophole for attackers to prepare an exploit. By the time developers release a patch (a small program to fix vulnerable files), the malware can do massive damage. All users are at risk, including the most cautious and attentive. The consequences of using an exploit can be very different. It depends on the task that malware posed: from disrupting the system to losing large sums of money, secret information. You can secure your device from exploit if you use anti-virus programs from well-known companies that are constantly improving their products. Regular updating of the operating system and application programs, refusal to click on suspicious links, ignoring spam messages, and careful attention to financial transactions will reduce the risk of infection [9].

Conclusion and prospects for further research

As stated above, exploits are a subtype of malware, but they are not detected by all security programs. Successful detection requires a defensive solution to use behavioral analysis - this is the only reliable method to combat exploits. Malicious programs can be numerous and varied, but most of them have similar behaviors.

Due to the fact that exploits are the result of perfect flaws, their elimination is the direct responsibility of the developers, so it is the authors who will have to prepare and send the bug fix. Nevertheless, the obligation to keep the installed programs updated and to install update packages in a timely manner so as not to give hackers a chance to take advantage of vulnerabilities lies entirely with the user of the program. One of the possible ways not to miss the latest updates is to use an application manager that will ensure that all installed programs are updated, or - even better - use the automatic update search and installation tool.

Rely on common sense and follow the basic rules for safe browsing on the Internet. Hackers can only exploit the vulnerability if they can gain access to your PC. Do not open attachments in suspicious messages or download files from unknown sources. Keep installed programs up-to-date and install security updates in a timely manner. If you want to simplify this task as much as possible, download Avast antivirus, which will not only provide reliable protection against all types of malware, but also help with installing the latest updates for third-party programs.

References

1. Andrianov V.I., Andronov A.V. Intelligent means of ensuring information security of automated systems in conditions of uncertainty // Journal of scientific publications of graduate students and doctoral students. 2010.No 8 (50). S. 120-121.
2. Vakhrushev I. A. et al. Method for searching for format string vulnerability // Transactions of the Institute for System Programming of the Russian Academy of Sciences, vol. 27, no. 4, 2015, pp. 23-38. DOI: 10.15514 / ISPRAS-2015-27 (4) -2.
3. Buynevich M.V., Izrailov K.E. Utility for searching for vulnerabilities in software of telecommunication devices using machine code algorithmization. Part 1. Functional architecture // Information Technologies and Communications. 2016.V. 4.No 1.P. 115-130.
4. Krasov A.V., Shterenberg S.I., Fakhrutdinov R. M., Ryzhakov D. V., Pestov I. E. Analysis of enterprise information security based on user data collection from open resources and monitoring of information resources using a machine training // T-comm: Telecommunications and transport. 2018.V. 12.No 10.P. 36-40.
5. Padaryan V.A., Kaushan V.V., Fedotov A.N. Automated method for constructing exploits for the buffer overflow vulnerability on the stack. Proceedings of ISP RAS, vol. 26, no. 3, pp. 127-144. DOI: 10.15514 / ISPRAS-2014-26 (3) -7.
6. Heelan S. Automatic generation of control flow hijacking exploits for software vulnerabilities. Master's thesis, University of Oxford, 2009.
7. Huang S.K. et al. Crax: Software crash analysis for automatic exploit generation by modeling attacks as symbolic continuations Software Security and Reliability (SERE), 2012 IEEE Sixth International Conference on. IEEE, 2012, pp. 78-87.
8. Fedotov A.N. A method for evaluating the exploitability of software defects. Proceedings of ISP RAS, vol. 28, no. 4, 2016, pp. 137-148. DOI: 10.15514 / ISPRAS- 2016-28 (4) -8.
9. Shterenberg S.I., Andrianov V.I. Investigation of adaptive attack techniques based on hidden attachment in executable files // Collection of articles of the International scientific and technical conference "Science, Technology, Innovations" (Bryansk, March 25–27, 2014) Bryansk: Reliable cars, 2014. S. 287-294.