

**Report on the topic:**

**" Vulnerability objects "**

|  |  |
| --- | --- |
|  | **Maid by Nurgaliev Devid**  **Group: 2ИСП11-15**  **Supervisour Klimova I.V.**  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |

**Moscow, 2021**

**Content**

Introduction.....................................................................………… 3

**Chapter 1. General characteristics of vulnerability** ...................4

1.1 Vulnerability in computer security.....................……...…........ 4

**Chapter 2. Objects of vulnerability**..............................….….......5

2.1 Identification of vulnerabilities................................................. 5

2.2 Objects vulnerabilities..................……………………..............6  
 Conclusion………………………………….…….….………….... 9

References………………………….....…........………………….10

**Introduction**

In any kind of activity, people tend to make unintentional mistakes, as a result of which they manifest themselves in the process of creating or using products or systems. In general, an error means a defect, error, or unintentional distortion of an object or process. It is assumed that the correct, reference state of the object is known, in relation to which the presence of a deviation — error can be determined.

With any fixed source data, programs are executed along certain routes and produce absolutely certain results, however, numerous variants of program execution with a variety of source data are presented to an external observer as random. In this regard, defects in the functioning of information systems that do not have malicious sources appear externally as random, have different nature and consequences. In particular, they can lead to catastrophic consequences corresponding to a violation of the security of the use of IP.



Fig. 1. A model for analyzing the security of information systems in the absence of malicious threats.

**Chapter 1. General characteristics of vulnerability**

* 1. Vulnerability in computer security

In computer security, the term "vulnerability" is used to denote a flaw in the system, using which it is possible to intentionally violate its integrity and cause incorrect operation. The vulnerability may be the result of programming errors, flaws made in the design of the system, unreliable passwords, viruses and other malware, script and SQL injections. Some vulnerabilities are known only theoretically, while others are actively used and have known exploits.

Usually, a vulnerability allows an attacker to "trick" an application — to perform actions unintended by the creator or force the application to perform an action for which it should not have rights. This is done by inserting data or code into the program in some way in such places that the program will perceive them as "its own". Some vulnerabilities appear due to insufficient verification of the data entered by the user, and allow you to insert arbitrary commands into the interpreted code (SQL injection). Other vulnerabilities appear due to more complex problems, such as writing data to a buffer without checking its boundaries (buffer overflow). Vulnerability search is sometimes called probing, for example, when they talk about probing a remote computer - they mean searching for open network ports and the presence of vulnerabilities related to applications using these ports.

The method of informing about vulnerabilities is one of the points of contention in the computer security community. Some experts advocate immediate full disclosure of vulnerabilities as soon as they are found. Others advise to report vulnerabilities only to those users who are most at risk, and to publish full information only after a delay or not to publish at all. Such delays may allow those who have been notified to correct the error by developing and applying patches, but they may also increase the risk for those who are not privy to the details.

There are tools that can help in detecting vulnerabilities in the system. Although these tools can provide the auditor with a good overview of possible vulnerabilities existing in the system, they cannot replace human involvement in their assessment.

To ensure the security and integrity of the system, it is necessary to constantly monitor it: install updates, use tools that help counteract possible attacks. Vulnerabilities were found in all major operating systems, including Microsoft Windows, Mac OS, various variants of UNIX (including GNU/Linux) and OpenVMS. Since new vulnerabilities are found continuously, the only way to reduce the likelihood of their use against the system is constant vigilance and the use of updated software versions.

**Chapter 2. Vulnerability objects**

2.1 Identifying vulnerabilities

To identify vulnerabilities, pentests are conducted, during which a list of systems to be tested and a specific goal are usually determined, and then the available information is analyzed and means are selected to achieve this goal. The purpose of the penetration test can be a "white box" (about which preliminary and system information is provided to the tester in advance) or a "black box" (about which only basic information is provided - if any - except for the company name).

When identifying (searching), analyzing and eliminating vulnerabilities in the ISPDn (personal data information system), the following should be carried out:

1. Identification (search) of vulnerabilities related to code errors in software (firmware) (system-wide, application, special), as well as information security software, the correct installation and configuration of information security tools, hardware and software, as well as the correctness of the operation of information security tools in their interaction with hardware and software;

2. Development of reports based on the results of identification (search) of vulnerabilities with a description of the identified vulnerabilities and a plan of measures to eliminate them;

3. Analysis of reports with the results of vulnerability search and assessment of sufficiency of implemented personal data protection measures;

4. Elimination of identified vulnerabilities, including by installing software updates of information security tools, system-wide software, application software or firmware of technical means;

5. Informing the officials of the personal data operator (users, administrators, information protection units) about the results of the vulnerability search and assessment of the sufficiency of implemented personal data protection measures.

The published data of developers of information security tools, system-wide, applied and special software, hardware, as well as other databases of vulnerabilities are used as sources of information about vulnerabilities.

Identification (search), analysis and elimination of vulnerabilities should be carried out at the stages of creation and operation of ISPDn. At the operational stage, the search and analysis of vulnerabilities is carried out at intervals established by the operator of personal data. At the same time, it is mandatory for critical vulnerabilities to search and analyze vulnerabilities in case of publication in publicly available sources of information about new vulnerabilities in information security tools, hardware and software used in ISPDn.

If it is impossible to eliminate the identified vulnerabilities by installing updates to the information security software, system-wide software, application software or firmware of technical means, it is necessary to take actions (configuring information security tools, changing the mode and procedure for using ISPDn) aimed at eliminating the possibility of using the identified vulnerabilities.

The operator must receive from trusted sources and install updates to the vulnerability database.

The rules and procedures for identifying, analyzing and eliminating vulnerabilities are regulated in the organizational and administrative documents of the personal data operator for the protection of personal data.

2.2 Vulnerability objects

Common types of vulnerabilities include:

Security violations of memory access, such as:

* Buffer overfills;
* Hanging pointers;

Input validation errors, such as:

* Formatting string errors;
* Onever support for interpreting command shell metacharacters;
* oSQL injection;
* Code Injection;
* E-mail Injection;
* Directory traversal;
* On-site scripting in web applications;
* On-site scripting in the presence of SQL injection;
* гонкиRace conditions such as:
* Time-of-check-by-time-of-use errors;
* Lights of symbolic links;
* привилегPrivilege confusion errors such as:
* Processing cross-site requests in web applications;

Escalation of privileges, such as:

* oShatter attack;
* нулZero-day vulnerability;

Memory access security is a concept in software development that aims to avoid software errors that lead to vulnerabilities related to accessing computer RAM, such as buffer overflows and dangling pointers.

Buffer overflow is a phenomenon that occurs when a computer program writes data outside of the buffer allocated in memory.

A dangling pointer or a dangling reference is a pointer that does not point to a valid object of the corresponding type. This is a special case of memory security violation.

SQL code injection is one of the most common ways of hacking websites and programs working with databases, based on the introduction of arbitrary SQL code into a query.

E-mail injection is an attack technique used to exploit mail servers and mail applications that construct IMAP/SMTP expressions from user input that is not checked properly. Depending on the type of operators used by the attacker, there are two types of injections: IMAP injection and SMTP injection.

XSS ("cross-site scripting") is a type of attack on web systems, consisting in the introduction of malicious code into the page issued by the web system (which will be executed on the user's computer when they open this page) and the interaction of this code with the attacker's web server. It is a kind of "Code Injection" attack.

SiXSS ("Cross-site scripting in the presence of SQL injection") is a type of attack on vulnerable interactive information systems on the web; the introduction of malicious scripts executed on a client computer into a page issued by the system by embedding code in SQL injection. As a rule, this vulnerability occurs on the client side, when there is output of printable fields by performing SQL injection.

A race condition is a design error of a multithreaded system or application, in which the operation of the system or application depends on the order in which parts of the code are executed. The error got its name from a similar error in the design of electronic circuits.

CSRF ("cross-site request forgery") is a type of attacks on website visitors using the shortcomings of the HTTP protocol. If the victim visits a site created by an attacker, a request is secretly sent on her behalf to another server (for example, to the server of a payment system) that performs some malicious operation (for example, transferring money to the attacker's account). To carry out this attack, the victim must be authenticated on the server to which the request is sent, and this request should not require any confirmation from the user, which cannot be ignored or forged by the attacking script.

Privilege escalation is the use of a computer bug, vulnerabilities, errors in the configuration of an operating system or software in order to increase the level of access to computing resources that are usually protected from the user. As a result, an application with more authority than the system administrator intended can perform unauthorized actions. "Privilege escalation" refers to a situation when a user of a computer system somehow increases his authority in this system (in other words: he got the opportunity to do what he could not do before).

The "shatter attack" is a software technology used by hackers to circumvent security restrictions between processes of a single session in the Microsoft Windows operating system. It relies on a flaw in the messaging system architecture and allows one application to inject arbitrary code into any other application or service running in the same session. As a result, unauthorized privilege escalation may occur.

0-day —zero day) is a term denoting unresolved vulnerabilities, as well as malware against which protective mechanisms have not yet been developed.

The term itself means that the developers had 0 days to fix the defect: the vulnerability or attack becomes publicly known before the release of bug fixes by the manufacturer (that is, the vulnerability can potentially be exploited on working copies of the application without the ability to protect against it).

**Conclusion**

Complete elimination of the listed threats is fundamentally impossible. The task is to identify the factors on which they depend, to create methods and means to reduce their impact on IP security, as well as to effectively allocate resources to ensure protection that is equally strong in relation to all negative impacts.

Modern advances in microelectronics have significantly reduced the impact of failures and failures of computing facilities on the security of the IC. However, staff errors, data distortions in telecommunication channels, as well as accidental (in case of equipment failures) and necessary configuration changes of computing facilities remain significant threats to the security of the IP. The negative impact of these factors can be significantly reduced by appropriate methods and means of protection in programs and data.

The development and application of the concept and standards of open systems contributes to the prevention of defects in complex, distributed IS. At the same time, it should be borne in mind that their use is associated with some contradictory trends in the nomenclature and magnitude of threats affecting the technological security of PS and DB. Mass transfer of programs and data to various hardware and operating platforms contributes to the spread of defects and undetected errors remaining in the portable components. However, portable components are usually more thoroughly tested and tested and thus have a higher quality than those created without a focus on portability. Standardization and deep formalized control of interfaces and protocols of interaction of IC components make it possible to create complex, distributed ICS of high reliability and security. Strict compliance and control of compliance with open systems standards (often automatically implemented by CASE-tools) is a highly effective method of preventing a number of classes of errors and improving the technological security of the IP.

**References**

1. Бабаш, А.В. Информационная безопасность. Лабораторный практикум: Учебное пособие / А.В. Бабаш, Е.К. Баранова, Ю.Н. Мельников. — М.: КноРус, 2016. — 136 c.

2. Гафнер, В.В. Информационная безопасность: Учебное пособие / В.В. Гафнер. — Рн/Д: Феникс, 2017. — 324 c.

3. Громов, Ю.Ю. Информационная безопасность и защита информации: Учебное пособие / Ю.Ю. Громов, В.О. Драчев, О.Г. Иванова. — Ст. Оскол: ТНТ, 2017. — 384 c.

4. Ефимова, Л.Л. Информационная безопасность детей. Российский и зарубежный опыт: Монография / Л.Л. Ефимова, С.А. Кочерга. — М.: ЮНИТИ-ДАНА, 2016. — 239 c.

5. Ефимова, Л.Л. Информационная безопасность детей. Российский и зарубежный опыт. Монография. Гриф УМЦ «Профессиональный учебник». Гриф НИИ образования и науки. / Л.Л. Ефимова, С.А. Кочерга. — М.: ЮНИТИ, 2016. — 239 c.

6. Запечников, С.В. Информационная безопасность открытых систем. В 2-х т. Т.1 — Угрозы, уязвимости, атаки и подходы к защите / С.В. Запечников, Н.Г Милославская. — М.: ГЛТ, 2017. — 536 c.

7. Запечников, С.В. Информационная безопасность открытых систем. В 2-х т. Т.2 — Средства защиты в сетях / С.В. Запечников, Н.Г. Милославская, А.И. Толстой, Д.В. Ушаков. — М.: ГЛТ, 2018. — 558 c.

8. Малюк, А.А. Информационная безопасность: концептуальные и методологические основы защиты информации / А.А. Малюк. — М.: ГЛТ, 2016. — 280 c.

9. Партыка, Т.Л. Информационная безопасность: Учебное пособие / Т.Л. Партыка, И.И. Попов. — М.: Форум, 2016. — 432 c.

10. Петров, С.В. Информационная безопасность: Учебное пособие / С.В. Петров, И.П. Слинькова, В.В. Гафнер. — М.: АРТА, 2016. — 296 c.

11. Семененко, В.А. Информационная безопасность: Учебное пособие / В.А. Семененко. — М.: МГИУ, 2017. — 277 c.

12. Чипига, А.Ф. Информационная безопасность автоматизированных систем / А.Ф. Чипига. — М.: Гелиос АРВ, 2017. — 336 c.