Applied Aspects of Security Testing

Alexander V. Dorofeev
Training Center Echelon, NPO
Moscow, Russia
mail@npo-echelon.com

Yrii V. Rautkin
Fedorova Institute of Applied Geophysics
Moscow, Russia
niira7@mail.ru

***Abstract*—The article is concerned with the procedure for ethical testing of computer systems' resource security. The capabilities of the Metasploit Framework testing tool are considered. The key "pain points" of modern corporate systems are identified. The general scheme and the security testing procedure are suggested. Vulnerability collection phases, checking the probability of exploiting discovered vulnerabilities, influence zone extension and privilege escalation are discussed in detail. It is shown that the proposed approach allows the maximum number of vulnerabilities to be detected.**

***Keywords*— information security; security testing, penetration testing, exploit, Metasploit Framework, Scanner-VS**

I. INTRODUCTION

Very few experts specializing in security testing have ever faced the situation when they were unable to fully compromise a network in the course of internal penetration testing [7]. The reasons for ethical hackers to succeed are trite: weak passwords, no critical security updates, configuration errors [11, 13]. This brings up the questions: if vulnerability causes are so trite, could they devise a list of key tests to be carried out independently by a system administrator and is there a tool that could help him do it? The tool of our choice is Metasploit Framework that can be installed by the user or advantage can be taken of what is offered by the Scanner VS complex [6, 10, 15, 16, 21]. Note that it is not our intention in this article to focus on testing security of web applications as these represent a separate testing area [2].

Before coming to grips with engineering problems, we should understand what may interest potential attackers and how they will act.

II. INFORMATION ATTACKERS ARE INTERESTED IN

As a rule, cyber-attackers are motivated by money earned on successful hacking (theft of funds, blackmail, carrying out orders for a party concerned) or by their own curiosity and willing to check what they are capable of. Attackers may aim at whatever permitting them either to steal or earn. Apart from any system operated by an organization, their aim may be folders on file servers and documents on users' workstations.

III. HIGH RISK AREAS

The main reason why somebody fails to update security or change a default password is a lack of responsible and proper control [12]. This reason allows us to identify the following pain points confirmed by our penetration testing practice [1, 4, 6, 8, 9, 14, 18-20, 23, 24].

- **Systems outsourced to third parties.** Normally, these are accounting, security and process systems. Contractors do not care about their security as this is the customer's area of responsibility rather than theirs. Administrators tend to dodge liability, being unable to understand the process.
- **Test and design environment.** Since these are not production systems, administrators will leave them in the hands of developers who may bring in vulnerabilities.
- **Conditionally isolated systems.** If a network is physically isolated and has no access to the Internet, "responsible" specialists may decide there is no point in installing antivirus software and other protection tools and, certainly, it is not worth updating anything.
- **Shared network locations.** File servers, shared folders on servers and workstations. This is where backup files, scripts with credentials and passwords, and system logs can be easily found.
- **Critical IT infrastructure elements.** The domain controller contains a credentials database and may systems support authorization by a domain credential. The domain controller, therefore, is the number-one priority for any attacker in the Windows network.

IV. METASPLOIT FRAMEWORK GENERAL HANDLING PROCEDURE

As we have chosen Metasploit Framework as a security testing tool, we need to describe the basic algorithm of handling its constituent modules.

The module is handled by performing the following steps:

1. Search for a suitable module using the **search** command or Google.
2. Select the module by the **use** command.
3. View the chosen module settings using **show options** or **show advanced** commands.
4. Specify a specific setting using the **set** command.
5. Set verbose output using the **set verbose true** command.
6. Run the module using the **run** command.

V. General security testing procedure

Security testing of information systems is often a creative process that, nevertheless, can and should be structured to obtain comparable and complete testing results [1, 4, 5, 8, 9].

The following sources provide a good description of security testing methodologies [3, 17, 18, 22]:

- Penetration Testing Execution Standard (PTES);
- Open Source Security Testing Methodology Manual (OSSTMM);
- Technical Guide to Information Security Testing and Assessment (NIST SP 800-115);
- OWASP Testing Guide.

PTES offers a detailed structure of the tasks to be tackled during security testing and exemplifies the use of various tools, while giving hardly any details of Metasploit Framework. OSSTM is largely intended for information security managers and contains a very restricted amount of technical information. NIST SP 800-115 was adopted in 2008 and only partially covers modern approaches to security testing. OWASP Testing Guide is only concerned with security testing of web applications and contains a detailed and structured description of testing methods and a variety of tool options.

The following diagram (figure 1) depicts a generalized structure of the proposed security testing process.

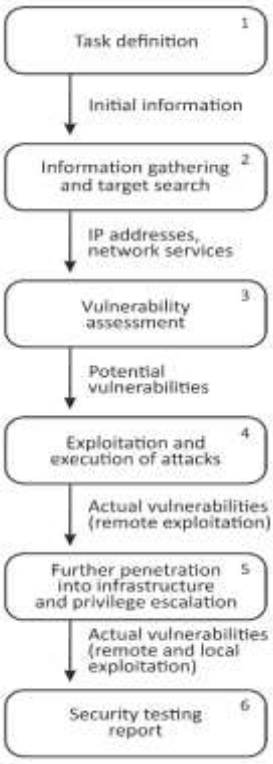

Fig. 1. Security testing flow chart

VI. Influence zone extension and privilege escalation

Phase 1. Task definition

Security testing of any IT infrastructures starts with task definition. In our case we will confine ourselves to searching for the maximum number of real vulnerabilities that may be exploited by potential attackers having physical access to an organization's computer network.

Phase 2. Information gathering and target search

In order to perform security testing, experts are provided with access to a company's network. In the course of preliminary information gathering, they will scan subsystems, identify computers' names, and find public network locations and critical resources.

Port scanning

Ports can be scanned using the "db_nmap – wrapping utility" command for nmap in Metasploit Framework, which allows scanning results to be saved to the database.

It should be borne in mind that if we do not specify the port range explicitly, 1,000 most commonly used ports will be scanned; if we specify the keys -F or -p-, 100 or 65,535 ports will be scanned, respectively.

What we can learn from port scanning results is not only which network ports are open, but also service versions (if the key "-sV" has been used) as well as the presumable OS version (key "-O") and the equipment manufacturer by the MAC address.

The completed port scanning shows which IP addresses hide domain controllers, DBMS servers, WEB, network equipment and workstations.

Search for public network locations

As discussed above, public network locations may contain a wealth of information useful for an attacker. It makes sense to search for such locations both with an anonymous credential (blank login/blank password) and a normal user's credential.

The "auxiliary/scanner/smb/smb_enumshares" and "auxiliary/scanner/nfs/nfsmount" modules should be used to search for SMB and NFS resources, respectively.

DBMS search

It is worth using the "auxiliary/scanner/mssql/mssql_ping" module to search for DBMS MS SQL as it helps not only find DBMS servers by the open UDP port 1434, but also identify the TCP port, via which the database is waiting to be connected.

NetBIOS name definition

It is often useful to define NetBIOS names (as they may also contain helpful information, for example, on which system a subsystem pertains to) by taking advantage of the auxiliary/scanner/netbios/nbname module.

Phase 3. Vulnerability Search

The table below lists the key vulnerability detection methods.

Table 1. The key vulnerability detection methods

N	Method	Type of detected vulnerabilities	Examples
1	Identifying vulnerabilities by product version	Published	Identifying a product version by the network service banner and browsing for information on the known vulnerabilities of the product
2	Exploitation attempt	Configuration errors, published vulnerabilities	Attempting to connect to the Windows system through a zero session and unloading the list of user credentials Starting an exploit against a network service without its prior analysis for conformity to the service Attempting to intercept traffic by means of ARP poisoning
3	Configuration analysis	Configuration errors, published vulnerabilities	Analyzing the Windows register contents
4	Reverse engineering	Zero-day vulnerability	Disassembling an executable file to study the logic of program execution and data handling
5	Source code analysis	Zero-day vulnerability	Searching the php code for fragments related to filtration of data entered by the user to get around filtration rules and introduce JavaScript code
6	Fuzzing	Zero-day vulnerability	Entry into a web form of SQL queries and analysis of received error messages

From this list, Metasploit Framework implements modules for the methods “Exploitation attempt”, “Fuzzing” and partially “Identifying vulnerabilities by product version”.

The reason why “Identifying vulnerabilities by product version” is not fully implemented in Metasploit Framework is because it primarily uses vulnerability scanners, such as one from Scanner VS, to automatically detect potential vulnerabilities. Note, however, that some exploitation modules in Metasploit Framework support the “check” method that can be used to identify a vulnerability before its exploitation.

Data on network service versions obtained at the port scanning phase is suitable for manual vulnerability analysis. A security tester generates Google search queries of the "service version +vulnerability +exploit” type to find pages of specialized resources describing vulnerabilities and exploits.

Metasploit Framework contains a set of fuzzing modules to execute protocols, such as dns, ftp, http, smb, smtp, ssh etc. These modules are available at auxiliary/fuzzers/.

It should be noted that, since security testing projects are normally restricted to a period of 2-3 weeks, security testers confine themselves to automated and manual search of vulnerabilities by version and to exploitation attempts.

Phase 4. Exploitation and execution of attacks

In order to exploit vulnerabilities, network services and applied software make use of exploits from Metasploit Framework exploit section. The current number of Metasploit Framework’s ready-to-use exploits is nearing 2,000.

In order to find suitable exploits, security testers utilize the “search” command by the CVE code, service name or version (for example, search vsftpd).

When exploiting a vulnerability, the so-called payload is to be specified. The payload is a code run on a compromised machine. There are a variety of payloads in Metasploit Framework, such as a remote command line, creating a credential, booting a remote administration system etc. Using the remote command line is often the best choice. Metasploit Framework has an extended command line version, Meterpreter, which is now particularly popular with testers.

Password brute forcing

Password brute forcing has been the most dangerous attack over decades. Metasploit Framework contains a lot of modules designed to execute such attacks. The table below lists the modules that experts typically come across in security testing.

Table 2. The testing modules

N	Protocol/ application	Module path
1	smb	auxiliary/scanner/smb/smb_login
2	ftp	auxiliary/scanner/ftp/anonymous (checking for anonymous entry possibility) auxiliary/scanner/ftp/ftp_login
3	ssh	auxiliary/scanner/ssh/ssh_login
4	telnet	auxiliary/scanner/telnet/telnet_login
5	postgresql	auxiliary/scanner/postgres/postgres_login
6	mysql	auxiliary/scanner/mysql/mysql_login
7	oracle	auxiliary/admin/oracle/oracle_login
8	tomcat	auxiliary/scanner/http/tomcat_mgr_login

A complete list of Metasploit Framework’s similar modules can be obtaining by typing the “search login” command.

Noteworthy is that most of the modules require specifying a list of credentials and verifiable passwords, but some of them already contain compiled lists of default values worth taking advantage of.

Metasploit Framework has modules for specific computer attacks. This article considers only the most typical ones.

ARP-poisoning

When executing such an attack, the attacker seeks to "poison" ARP tables of two subsystems, the traffic between which he wants to intercept. An attack is often undertaken against the workstation of a particular user (system administrator, chief accountant etc.) and a domain controller or router. Once ARP tables are poisoned, both victim subsystems share network packets via the attacker's computer. Having run a sniffer, the attacker captures the data of interest, for example, sessions of authentication with password hashes.

An ARP poisoning attack in Metasploit Framework can be executed by making use of the "auxiliary/spoof/arp/arp_poisoning" module.

Pass-the-hash

Successful authorization when executing the NTLM protocol does not require knowing the password – it is enough to have the password hash and credential name. Any operating system using the NTLM protocol can be susceptible to this vulnerability.

A pass-the-hash attack can be executed using the "exploit/windows/smb/psexec" module.

This security testing phase provides us with a list of vulnerabilities that can be exploited by attackers remotely. The exploits run and attacks executed have provided us with access to various systems and with information about compromised credentials.

Testers collect screenshots confirming access as evidence of successful penetration.

Phase 5. Influence zone extension and privilege escalation

The existing access to a system often allows it to be extended to other systems. Privilege escalation permitting a normal user to become an administrator is also possible sometimes.

Let us consider two standard situations that a tester should be aware of to make security testing easier.

Lazy users making use of identical passwords

Users like utilizing identical passwords in different systems, so it is worth checking once selected pairs "login:password" in all accessible systems.

Lazy administrators forgetting to delete critical data from the test environment

Serious systems implemented by major companies normally have a test environment used to try out modifications, train users etc. Test environments are often created by restoring from production backups. Because they are test environments, administrator sometimes fail to pay due attention to information security issues. For example, they may create an administrator's credential with an easily guessed password or fail to set critical OS updates. Upon receiving access to a test environment, security testers unload user data (logins/password hashes) that are largely consistent with those employed in a production system.

Post-exploitation modules in Metasploit Framework

Metasploit Framework has a set of so-called post-exploitation modules designed to perform the following tasks for access extension and privilege escalation:

- Searching for suitable local exploits (post/multi/recon/local_exploit_suggester);
- Running a keylogger (post/windows/capture/keylog_recorder);
- Gathering credentials and hashes (post/windows/gather/credentials/credential_collector) etc.

With this step performed, security testers obtain maximum access and pinpoint actual local vulnerabilities.

Phase 6. Report development

The outcome of security testing is a report on discovered vulnerabilities. The report's key component is information on vulnerabilities, which is normally provided to the customer in the structured form:

- Detection – information on vulnerability name and codes and a list of vulnerability-prone subsystems.
- Exploitation – screenshots and logs demonstrating vulnerability exploitation;
- Risk – what vulnerability exploitation may result in;
- Recommendations – technical and organizational recommendations on elimination of vulnerabilities.

Since Metasploit Framework has no security testing report generation feature, the report is developed by testers.

VII. CONCLUSION

We have considered a comprehensive approach to security testing, which can be implemented through Metasploit Framework. Metasploit Framework is an aid in completing the key phases of security testing, except for automated vulnerability search and report generation. These phases, however, are implemented in the Scanner VS complex that comprises Metasploit Framework. When used in conjunction, the described methodology, Scanner VS and Metasploit Framework help discover the maximum number of actual vulnerabilities.

REFERENCES

[1] Scaner-VS. http://scaner-vs.ru/trial/

[2] Abraham K White. Hacking: The Underground Guide to Computer Hacking, Including Wireless Networks, Security, Windows, Kali Linux and Penetration Testing. – CreateSpace Independent Publishing Platform, 2017. 230 p.

[3] Barabanov A.V., Lavrov A.I., Markov A.S., Polotnyanschikov I.A., Tsirlov V.L. The study into cross-site request forgery attacks within the framework of analysis of software vulnerabilities. Trudy ISP RAN/Proc. ISP RAS, vol. 29, issue 5, 2017, pp. 7-18. DOI: 10.15514/ISPRAS-2017-29(5)-1.

[4] Chris McNab. Network Security Assessment: Know Your Network. – O'Reilly Media, 2017. 508 p.

[5] Corey P. Schultz, Bob Perciaccante. Kali Linux Cookbook - Second Edition: Effective penetration testing solutions. – Packt Publishing, 2017. 438 p.

[6] Daniel W. Dieterle. Basic Security Testing with Kali Linux 2. – CreateSpace Independent Publishing Platform, 2016. 380 p.

[7] David Kennedy, Jim O'Gorman, Devon Kearns. Metasploit: The Penetration Tester's Guide. – No Starch Press, 2011. 328 p.

[8] Dorofeev A. Preparing for CISSP: telecommunications and network security. Voprosy kiberbezopasnosti *[Cybersecurity issues]*, 2014, No 4(7). P. 69-74. (In Rus).

[9] Evan Lane. Hacking with Python: Beginner's Guide to Ethical Hacking, Basic Security, Penetration Testing, and Python Hacking. – CreateSpace Independent Publishing Platform, 2017. 106 p.

[10] Georgia Weidman. Penetration Testing: A Hands-On Introduction to Hacking. – No Starch Press, 2014. 528 p.

[11] Jessey Bullock, Jeff T. Parker. Wireshark for Security Professionals: Using Wireshark and the Metasploit Framework. – John Wiley & Sons, 2017. 288 p.

[12] Josh Thompsons. Hacking: Hacking For Beginners Guide On How To Hack, Computer Hacking, And The Basics Of Ethical Hacking. – CreateSpace Independent Publishing Platform, 2017. 112 p.

[13] Joshua Picolet. Hash Crack: Password Cracking Manual (v2.0). CreateSpace Independent Publishing Platform, 2017. 112 p.

[14] Matt Walker. CEH Certified Ethical Hacker All-in-One Exam Guide, Third Edition. – Oracle Press, 2016. 525 p.

[15] Michael Hixon, Justin Hutchens. Kali Linux Network Scanning Cookbook - Second Edition: A Step-by-Step Guide Leveraging Custom Scripts and Integrated Tools in Kali Linux. – Packt Publishing, 2017. 634 p.

[16] Nipun Jaswal. Metasploit Bootcamp: The fastest way to learn Metasploit Paperback. – Packt Publishing, 2017. 230 p.

[17] Open Source Security Testing Methodology Manual. Online http://www.isecom.org/research/

[18] OWASP Testing Guide. Online https://www.owasp.org/index.php/OWASP_Testing_Guide_v4_Table_of_Contents

[19] Penetration testing execution standard. Online http://www.pentest-standard.org

[20] Peter Kim. The Hacker Playbook 2: Practical Guide to Penetration Testing. – CreateSpace Independent Publishing Platform, 2015. 358 p.

[21] Raphael Hertzog, Mati Aharoni, Jim O'Gorman. Kali Linux Revealed: Mastering the Penetration Testing Distribution. – Offsec Press, 2017. 314 p.

[22] Scaner-VS. Online http://scaner-vs.ru/trial/

[23] Technical Guide to Information Security Testing and Assessment. Online http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-115.pdf

[24] Tedi Heriyanto, Lee Allen. Kali Linux – Assuring Security by Penetration Testing. – Packt Publishing, 2014. 454 p.

[25] Wil Allsopp. Advanced Penetration Testing: Hacking the World's Most Secure Networks. – John Wiley & Sons, 2017. 288 p..