УДК 004.451, 004. 492

Я. Я. Стефінко, А. З. Піскозуб, Р. І. Банах
Національний університет "Львівська політехніка",
кафедра безпеки інформаційних технологій,
кафедра захисту інформації

ТЕСТУВАННЯ НА ПРОНИКНЕННЯ З METASPLOIT І SHELL СКРИПТАМИ

Наведено інформацію про загрози в комп'ютерних мережах і системах, і один з шляхів захистити їх – тестування на проникнення (пентест). Найпридатнішими інструментами для цієї цілі є ОС Kali Linux та його можливості роботи зі скриптами. В статті описуються методи і шляхи імплементації цих скриптів для успішного тестування на проникнення. Проаналізовано сучасне програмне забезпечення для пентестів, наведена інформація про інтеграцію скриптів з програмою Metasploit і продемонстровано приклади окремих скриптів у Kali Linux.

Ключові слова: проактивний захист, тест на проникнення, скрипт, вразливості, зловмисник, захищеність.

This article contains information about security threats in computer networks and systems, and one of the ways to protect it is penetration testing. Most useful tools for this purpose are OS Kali Linux and its shell scripts. We describe the methods and ways of implementation of these scripts to assist us in successful pentest. We analyze the current free pentestsoftware, integration of Metasploit and shell scripts and demonstrate examples for using special bash scripts of OS Kali Linux.

Key words: Kali Linux, Metasploitable, pentest, penetration, scripts, shell, bash, vulnerability, Metasploit Framework, pentest, security, proactive defense, corporate networks.

Вступ

У світі інформаційних технологій комп'ютерні мережі і системи стають невіддільним інструментом в житті сучасної людини, і все більшу роль при цьому відіграють аспекти інформаційної безпеки. Разом з тим кіберзлочинність зростає, кількість вразливих ОС та іншого ПЗ також невпинно росте. Зловмисники постійно поповнюють свій арсенал все новими хакерськими програмами, вірусами, троянами тощо. Слідом за цим неминуче з'являються нові методи і способи для захисту комп'ютерних систем.

Технологія тестування на проникнення сьогодні рясніє спрощеними графічними інтерфейсами для користувача. Незважаючи на простоту використання, вони часто пропонують дуже мало контролю над операціями і не пропонують інформативного досвіду для своїх користувачів. Ще одним недоліком є те, що багато з цих методів оцінювання безпеки розроблено тільки для ідентифікації та автоматизації експлуатації у найочевидніших і традиційних випадках вразливостей. Для будь-якого іншого практичного прикладу вразливості пентестеру потрібно покладатися на свої власні сценарії та інструменти оцінювання.

Тестування на проникнення і скрипти

Основний набір навичок доброго пентестера проникнення передбачає наявність щонайменше, елементарних навичок в написанні сценарії або в використанні мов програмування – таких, як Python, Ruby, Perl тощо. Це пояснюється тим, що вони можуть впоратися з особливими і винятковими екземплярами вразливостей з їх спеціальними засобами, і здатні до автоматизації тестування безпеки, що значно спростить дослідження [3].

Тест на проникнення (пентест) дозволяє моделювати несанкціонований доступ в інформаційні системи, а також інші дії, які дозволяють порушити нормальне функціонування систем і бізнес-процесів. По суті це метод оцінювання захищеності інформаційних систем та об'єктів від несанкціонованого використання [6].

Командні оболонки, зокрема Bourne Again Shell (bash) є, можливо, одними з найважливіших з погляду здійснення пентесту. Без багатьох утиліт оболонки bash і потенціалом, що дається користувачам шляхом об'єднання і взаємодії системних утиліт в програмований спосіб (так звані Bash сценаріїв чи скрипти), багато з важливих проблем безпеки в сучасному світі було б дуже складно вирішити [3]. Утиліти, такі як grep, wget, vi, і awk дозволяють своїм користувачам здійснювати дуже потужну обробку рядків, видобування даних і управління інформацією. Системні адміністратори, розробники, інженери безпеки, тестери протягом багатьох років покладаються на цей потенціал вирішення щоденних технічних проблем і ефективність.

Принципово оболонка bash є найбільш вживаною і стандартизованою. Це означає, що можна гарантувати певний базовий набір поведінки для bash-скриптів або набору команд незалежно від операційної системи і реалізації bash. Загальну структуру і синтаксис Linux/Unix часто буває важко оцінити початківцям, можливо, через відсутність підказок, натяків і зручного графічного інтерфейсу та ззовнішньої привабливості.

Середовище bash буде представлене тут на прикладі спеціалізованої операційної системи Kali Linux. Калі – це дистрибутив, взятий з Debian, упакований з утилітами, які орієнтовані виключно на вирішення технічних проблем безпеки і тестування на проникнення.

Shell як універсальне розв'язання задач пентесту у поєднанні з Metasploit

Наведемо приклади найкорисніших команд bash для ефективного виконання наших подальших технічних задач в Kali Linux. По-перше, для навігації системою чи файлами ми використовуємо *cd, pwd, ls, find, man, grep.* Нами протестовано, що перенаправлення вводу-виводу(I/O) дозволяє побачити всі результати роботи скриптів і краще зрозуміти процеси. Наприклад, нами випробовувано можливості автоматизації в Linux, а саме через вивід від *nmap* або *tcpdump,* або *кейлоггерів* (утиліт для захоплення натиснутих клавіш), шляхом подання інформації на *output* в інший файл або програму для подальшого аналізу. Для перенаправлення виводу потрібно тільки додати в кінці команди символ >, а для введення з файла чи програми – навпаки <. Для аналогічних цілей обміну виведенням між процесами використовувалось *pipe,* тобто | [4].

Також ми проаналізували, як інтегрувати утиліти, такі як: *nmap, whois, dig* та інші мережеві "швейцарські ножі", щоб дослідити стан безпеки на конкретних хостах чи в певних локальних мережах.

Whois servers містять інформацію про IP адреси, доменні назви та іншу відповідну інформацію про певні організації, якими ми можемо зацікавитися під час пентесту. За допомогою команди *dig* ми отримували всю можливу інформацію про певний домен чи IP адресу із всесвітньої павутини. Для отримання конкретнішої інформації також часто використовують *dnsmap і dnsenum.*

Для оглядання і окреслення цільового хоста ми використовували Network mapper (*nmap*) та *arping*. *Nmap* став де-факто стандартом для мережевого оцінювання і може робити значно більше ніж *hping, fping, і arping.* У певних випадках часто для оцінювання міжмережевих екранів ми, як пентестери використали більш налаштовувані менеджери мережевих пакетів у різних протоколах мережевого рівня. Саме тут і пригодяться *hping, fping,* і *arping* [2].

Ось приклад ICMP сканування з Nmap:

nmap –sn –v –-reason 192.168.10.0/24

Metasploit – це найрозширеніша сьогодні платформа для тестування на проникнення та розроблення експлойтів у всьому світі [1]. Цієї утиліти необхідно і достатньо для тестування, пошуку і розроблення експлойтів для вразливостей ОС чи додатків. Основним і універсальним середовищем виконання для Metasploit є утиліта з bash – ***msfcli*** [5]:

msfcli [MODULE] [OPTIONS] [MODE]

[MODULE] := [exploit/ | auxiliary/* | payload/* | post/*]*

*[OPTION] := [[option_name] = [value] <space>]**

[MODE] := [A | AC | C | E | H | I | O | P | S | T]

Linux і bash скрипти дають нам змогу дуже вдало комбінувати деякі команди, щоб отримати дуже зручний вивід даних з застосуванням до них, певних можливостей Metasploit. Наприклад, використаємо MSFcli, Nmap та awk [3]:

for ip in `nmap –v –T5 –p[PORT] [HOST] | awk –F\ '/[PORT]\/[tcp|udp]

on/ { print $6 }'`; do msfcli [MODULE] RHOST=$ip E; done

Також до пакета утиліт Metasploit входять msfconsole, msfpayload, meterpreter тощо. Важливими утилітами bash, якими ми оперували на етапі експлуатації, в процесі пентесту є також *arpspoof, macchanger, tcpdump, ettercap, sslyze, w3af, arachni, sqlmap, john-the-ripper і smtpwalk*[3].

Наведемо один з ефективних прикладів використання скриптів у поєднанні з платформою Metasploit Framework:

for ip in `nmap –v –T5 –p80 [HOST] |

awk –F\ '/80\/tcp on/ { print $6 }'`;

do msfcli auxiliary/fuzzers/http/http RHOST=$ip E; done

Авторами також проаналізовано, що bash у поєднанні з Ruby скриптами, якими можна управляти Metasploit Framework зсередини, можна автоматизувати та вдосконалити тестування на проникнення. Ми дослідили такий спосіб комбінування скриптами та конфігураціями Metasploit Framework, який дозволить нам в майбутньому автоматизувати певні етапи в процесі пентестування [5]. Для дослідження використано офіційно задокументовані можливості програмувати на Ruby на платформі Metasploit. Зразок скрипту наведено на рисунку.

```
use auxiliary/scanner/mssql/mssql_login
set USER_FILE /opt/sql_brute/sql_users.txt
set PASS_FILE /opt/sql_brute/sql_wordlist.txt
set VERBOSE false
set THREADS 255
<ruby>
framework.db.hosts.each do |host|
  host.services.each do |service|
    if service.name == "mssql" and service.state == "open"
      self.run_single("set RHOSTS #{host.address}")
      self.run_single("set RPORT #{service.port}")
      self.run_single("run")
    end
  end
end
</ruby>
```

Скрипт для автоматизації тестування на проникнення

На рисунку видно перебір логінів і паролів для проникнення в SQL бази з використанням автоматизації в оболонці ОС Kali Linux. Тобто так ми можемо вдало поєднувати ці два інструменти для підвищення ефективності виконання і для автоматизації деяких процесів у межах пентесту.

Методи тестування на проникнення постійно удосконалюються і, на жаль, використовуються не лише в оборонних, але і в наступальних цілях. Саме тому важливим є питання своєчасного виявлення вразливостей у захищених чи стратегічно важливих для держави системах, періодично проводячи тестування на проникнення.

Висновки

Сьогодні щоденно виявляють нові вразливості у всесвітньо відомих і поширениз протоколах і системах (Bash shellshock, SSL heartbleed і т.д.). Тому жодну систему чи протокол зараз не можна вважати цілком і абсолютно захищеними. Тому виникає необхідність проведення пентестів, які у поєднанні з різними скриптовими мовами дадуть змогу сповна використати усі можливості ОС Kali Linux. Отож, ми можемо стверджувати, що зараз скриптинг та Metasploit Framework взагалі є ключовими інструментами для здійснення ефективних тестів на проникнення та для виявлення нових вразливостей, адже вони дають змогу протестувати та дослідити детальніше мережеві протоколи та операційні системи загалом. Дослідження допоможуть нам ретельно розібрати, додати власні скрипти та структурувати процес тестування на проникнення для його оптимізації, автоматизації та покращення результатів у звітах. Своєю чергою, призведе до покращення захищеності досліджуваних систем, адже ми зможемо виявляти вразливості, випередивши зловмисників. Тобто, ми і надалі можемо продовжувати дослідження способів автоматизації для підвищення ефективності пентесту.

1. Піскозуб А. З. Використання тестування на проникнення в комп'ютерні мережі та системи для підняття їх рівня захищеності // Матеріали третьої міжнар. наук.-практ. конференції FOSS Lviv 2013. – Львів, 2013. 2. Стефінко Я. Я., Піскозуб А. З. Використання Kali linux та Metasploitable для тестування на проникнення в навчальних цілях // ІІІ Міжнародна науково-технічна конференція "Захист інформації і безпека інформаційних систем". – Львів, 2014. 3. Kennedy D., O'Gorman J. Metasploit. The penetration tester's guide. – No starch press, San Francisco, 2011. 332 с. 4. Keith Makan. Penetration Testing with the Bash shell. Birmingham – Mumbai, Packt Publishing, 2014, 151с. 5. Jason Andress, Ryan Linn. Coding for Penetration Testers. Elsiever – London, 2012, 321с. 6. Стефінко Я. Я., Піскозуб А. З. Використання відкритих операційних систем для тестування на проникнення в навчальних цілях // Вісник Нац. ун-ту "Львівська політехніка" "Комп'ютерні системи і мережі". – 2014.