**Лабораторная работа № 2 – Исследование подходов, применяемых при эксплуатации уязвимостей**

**Цель работы**

Изучение способов эксплуатации уязвимостей и механизмов противодействия.

**Формулировка задания**

При проведении исследования безопасности программного обеспечения одним из этапов работы является проверка найденных ошибок на возможность их эксплуатации. Для этого осуществляется написание, так называемых proof-of-concept эксплойтов (POC), демонстрирующих возможность эксплуатации найденной уязвимости.

Одной из наиболее распространенных ошибок является переполнение буфера. Оно может происходить как на стеке (stack overflow), так и на куче (heap overflow).

В рамках лабораторной работы необходимо осуществить эксплуатацию уязвимости, найденной в исполняемом файле в лабораторной работе №1. Необходимо реализовать полезную нагрузку в соответствии с указанным вариантом задания, которая будет выполняться в результате эксплуатации уязвимости.

В ходе выполнения лабораторной работы необходимо выполнить следующие действия:

1. Использовать полученные от преподавателя файлы для лабораторной работы №1 в соответствии со своим вариантом.
2. Получить вариант задания, в котором описан функционал полезной нагрузки.
3. Изучить защитные механизмы, затрудняющие эксплуатацию уязвимостей (DEP, ASLR, GS, Security Cookies, SMEP, SAFESEH, SEHOP и другие).
4. Изучить техники эксплуатации уязвимостей и способы обхода защитных механизмов (ret2libc, ROP, Stack Pivoting, Heap Spray и другие).
5. Используя скрипт mona.py для отладчиков Immunity Debugger/WinDBG или другой с аналогичной функциональностью осуществить получение списка загруженных в процесс модулей, не использующих технологию рандомизации адресов. Реализовать скрипт для отладчика.
6. Найти в динамических библиотеках без рандомизации адресов опкоды инструкций, необходимых для передачи управления (call esp, jmp esp, …) на полезную нагрузку. Реализовать для этих целей скрипт для отладчика (Immunity Debugger/WinDBG/IDA Pro).
7. Провести эксплуатацию найденной уязвимости, с использованием полезной нагрузки (shellcode) в соответствии с выданным вариантом.
8. Эксплуатацию уязвимости осуществлять под ОС семейства Windows версии не ниже Windows 7 x64. При эксплуатации запрещено изменение исполняемого файла, запуск сторонних исполняемых файлов, либо выполнение консольных команд. Полезная нагрузка должна быть написана на языке ассемблера, либо на языке C с использованием приемом разработки базонезависимого кода.
9. В отчете необходимо привести следующую информацию:

* перечень динамических библиотек, не использующих рандомизацию адресов;
* код, осуществляющий эксплуатацию уязвимости (exploit) с комментариями;
* схема эксплуатации уязвимости (описание стека при эксплуатации, описание приемов обхода защиты);
* формат файла, используемого для эксплуатации, с указанием значения полей, приводящих к исполнению кода;
* код полезной нагрузки в бинарном виде с комментариями;
* код полезной нагрузки на языке ассемблера / языке C с комментариями.

Ссылки

1. Gray\_Hat\_Python <https://www.reverse4you.org/translate/GHP/Book/Gray_Hat_Python.pdf>
2. Practical Malware Analysis

<http://venom630.free.fr/pdf/Practical_Malware_Analysis.pdf>

1. Exploit writing tutorial

<https://www.corelan.be/index.php/2009/07/19/exploit-writing-tutorial-part-1-stack-based-overflows/>

1. Exploit writing tutorial. Частичный перевод

<https://forum.reverse4you.org/showthread.php?t=1407>

1. Understanding Windows Shellcode

<http://nologin.org/Downloads/Papers/win32-shellcode.pdf>

1. Writing Shellcode with a C Compiler

<https://nickharbour.wordpress.com/>

<https://forum.reverse4you.org/showthread.php?t=2154>

<https://forum.reverse4you.org/showthread.php?t=2155>

Таблица 1 – Варианты полезной нагрузки к лабораторной работе №2

| **Вариант** | **Название прикладного программного обеспечения** |
| --- | --- |
| 1 | Открытие на прослушивание TCP-порта 8080 для выполнения команд атакующего (**TCP bind shell**) |
| 2 | Вывод содержимого каталога Windows в некоторый файл |
| 3 | Вывод списка всех дисков, которые есть в системе |
| 4 | Сохранение в файл списка открытых портов |
| 5 | Удаление из заданного каталога всех файлов, начинающихся на последовательность букв "dw" |
| 6 | Остановка указанного запущенного процесса |
| 7 | Удаление из реестра записи о драйвере Beep в CurrentControlSet |
| 8 | Вывод информации о процессоре |
| 9 | Сохранение в файл списка сетевых интерфейсов |
| 10 | Добавление программы в автозагрузку |
| 11 | Вывод содержимого каталога Program Files в некоторый файл |
| 12 | Получение и вывод списка загруженных в процесс браузера библиотек |
| 13 | Вывод списка процессов и их разрядности |
| 14 | Сохранение имени текущего пользователя в реестр |
| 15 | Вывод MessageBox'ом текущей даты |
| 16 | Подключение к порту компьютера атакующего (**Reverse TCP bind shell**) |
| 17 | Сохранение в файл списка запущенных сервисов |
| 18 | Вывод списка всех дисков, которые есть в системе |
| 19 | Подключение к порту компьютера атакующего (**Reverse TCP bind shell**) |
| 20 | Сохранение в файл списка сетевых интерфейсов |
| 21 | Сохранение в файл информации о свободном месте на диске |
| 22 | Внедрение кода в адресное пространство заданного процесса (inject) |
| 23 | Открытие на прослушивание UDP-порта 8081 для выполнения команд атакующего (UDP **bind shell**) |
| 24 | Получение и вывод списка загруженных текущим процессом библиотек |
| 25 | Сохранение в файл IP-адресов сетевых интерфейсов |
| 26 | Вывод MessageBox'ом время, прошедшее с момента запуска системы |
| 27 | Сохранение в файл информации о свободном месте на диске |
| 28 | Сохранение в файл списка сетевых интерфейсов |
| 29 | Сохранение в файл копии MBR (Master Boot Record) |
| 30 | Сохранение в файл списка сетевых интерфейсов |
| 31 | Внедрение кода в адресное пространство заданного процесса (inject) |
| 32 | Остановка указанного запущенного процесса |
| 33 | Удаление из заданного каталога всех файлов, имена которых заканчиваются на букву "d" |
| 34 | Остановка указанного запущенного процесса |
| 35 | Вывод MessageBox'ом время, прошедшее с момента запуска системы |
| 36 | Сохранение в файл информации о размере виртуальной памяти |
| 37 | Сохранение имени текущего пользователя в заданный файл |
| 38 | Добавление программы в автозагрузку |
| 39 | Вывод содержимого заданного каталога в некоторый файл |
| 40 | Вывод списка всех дисков, которые есть в системе |
| 41 | Подключение к порту компьютера атакующего (**Reverse TCP bind shell**) |
| 42 | Сохранение в файл списка открытых портов |
| 43 | Вывод списка приложений из автозагрузки |
| 44 | Отправка письма на некоторый ящик |
| 45 | Получение и вывод списка загруженных в процесс браузера библиотек |
| 46 | Сохранение в файл информации о размере оперативной памяти |
| 47 | Сохранение информации в реестр о размере оперативной памяти |
| 48 | Вывод информации о процессе (разрядность, ASLR) |
| 49 | Получение и вывод списка загруженных в процесс браузера библиотек |
| 50 | Создание в заданном каталоге файла, содержащего список содержимого каталога |
| 51 | Подключение к порту компьютера атакующего (**Reverse TCP bind shell**) |
| 52 | Сохранение в файл информации о свободном месте на диске |
| 53 | Открытие на прослушивание TCP-порта 8052 для выполнения команд атакующего (**TCP bind shell**) |