Лабораторная №1

Переполнение буфера

Цель работы:

Получить базовое представление о том, что такое уязвимость ПО.

Задача:

Разработать сервер на языке C/C++. Сервер должен принимать запрос в виде последовательности байт и возвращать на неё ответ, который зависит от входа (это может быть что угодно, например, контрольная сумма). Сервер должен содержать в себе уязвимость переполнения буфера. Необходимо продемонстрировать эксплуатацию уязвимости удалённого исполнения кода (RCE – remote code execution) (допустимо отключить все необходимые механизмы защиты). Также необходимо «закрыть» уязвимость двумя способами:

1. разработать вторую версию этого сервера, с исправленной уязвимостью в коде программы;
2. собрать уязвимый сервер со средствами противодействия эксплуатации (exploit mitigation).

Работу сервера можно демонстрировать при помощи команд netcat/telnet/wget/curl/…

Что нужно для начала работы:

1. компилятор C/C++ кода (cl или clang под Windows, gcc или clang под Linux/macOS);
2. отладчик (WinDbd (https://eax.me/windbg/) или x64dbg под Windows, gdb (https://eax.me/gdb/) или radare2 под Linux/macOS).

Что нужно прочитать, чтобы начать работать

1. DEP (data execution prevention);
2. ASLR (address space layer randomization);
3. stack-protector в gcc/Buffer Security Check в cl.exe;
4. как это делалось в 1996 году: http://phrack.org/issues/49/14.html

Приблизительный вид результата:

1. сервер, принимающий по TCP последовательность байт и возвращающий в ответе хэш от неё;
2. в коде сервера существует выделение буфера на стеке и его заполнение без контроля выхода за пределы;
3. набор команд для отключения необходимых систем защиты (DEP, ASLR, …) и команда для сборки сервера в виде, при котором возможна эксплуатация уязвимости удалённого выполнения кода;
4. команда netcat для демонстрации корректной работы сервера;
5. команда для сборки сервера с защитой от уязвимости;
6. сервер с исправленной уязвимостью и команда для его сборки.

Итого:

1. исходный код сервера с уязвимостью;
2. исходный код сервера без уязвимости;
3. команда для сборки сервера с уязвимостью без средств защиты;
4. команда для сборки сервера с уязвимостью с средствами защиты;
5. команда для демонстрации работы корректной работы сервера;
6. команда для демонстрации работы RCE уязвимости.