**Фахрутдинов Булат Загфярович**

**11-902**

**ЛАБОРАТОРНАЯ РАБОТА №1. LIME И VOLATILITY. ФОРЕНЗИКА.**

**Цели и задачи:** научиться искать и применять уязвимости ядра OC Linux при помощи инструментов захвата RAM памяти LiME и Volatility Framework.

**Ход работ:**

1. Был скачан образ OC Linux с разрешением «.vmdk» и установлен на виртуальную машину VMWare
2. Были проверены технические характеристики виртуальной машины: архитектура процессора – х64, количество процессоров – 1, объем ОЗУ – 512 MB, размер жесткого диска – 10 GB, тип сетевого подключения – сетевой мост.
3. Был выполнен вход в систему и запрос ip-адреса
   1. Введены логин и пароль: “msfadmin” для входа в систему
   2. Выполнена команда “sudo su” – для входа в режим суперпользователя и повторно введен пароль от пользователя ‘msfadmin’
   3. Выполнена команда “ip a” для просмотра адреса виртуальной ОС в сети. Адрес был локальным, поэтому было необходимо получить DHCP адрес, для связи с хостовым ПК
      1. Выполнена команда “dhclient” – для получения публичного адреса
   4. Адрес был записан для дальнейшей работы с хостовым ПК
4. Создание backup-файла репозиториев
   1. Выполнил команду “cd /etc/apt”
   2. Выполнил команду “cp sources.list sources.list.BKP;”
   3. Убедился в том, что файл успешно создан, выполнив команду “ls –l sources.list\*”
5. Просмотр, изменение и обновление списка репозиториев
   1. Так как новые репозитории не поддерживается данной версии ОС, необходимо изменить список репозиториев выполнив команду “vim sources.list”
   2. В текстовом редакторе выполняем последовательность команд: “двоеточие” – “%s/us.archive/old-releases/” – “Enter”
   3. Выходим из текстового редактора последовательностью команд: “двоеточие” – “wq” – “Enter”
   4. Выполняем “apt-get update” для обновления
6. Установка заголовочных файлов ядра
   1. Выполняем команду “uname -r”
   2. Выполняем последовательность команд: “apt-get install linux-headers-$(uname -r)” – “Y”
7. Скачивание и установка архива с исходными текстами LiME
   1. На хостовой машине скачиваем LiME из репозитория GitHUB
   2. На хостовой машине выполняется команда “scp ‘путь до файла на хостовой машине’ msfadmin@’ip-адрес полученный ранее’:/var/tmp/”
   3. Переходим в директорию ‘/var/tmp’ после чего выполняем распаковку архива командой “unzip -x LiME-master.zip”
   4. Выполняем последовательность команд: “cd LiME-master/src” – “make”
8. Скачивание и установка библиотеки libdwarf
   1. Аналогичные действия с п.7 (единственным различием будет распаковка командой “tar xf libdwarf-20140208.tar.gz”), до подпункта 7.4
   2. Переходим в папку ‘cd /var/tmp/dwarf-20140208’
   3. Выполняем последовательность команд: “./configure” – “make” для компиляции
   4. Перемещаем dwarf в исполняемые файлы командой: “cp dwarfdump/dwarfdump /usr/bin”
9. Скачивание и установка Volatility
   1. По аналогии с предыдущими пунктами выполняем последовательность действий: распаковка командой – “tar zxvf volatility-2.3.1.tar.gz”; переход в каталог ‘/var/tmp/volatility-2.3.1/tools/linux’ и генерация путем выполнения “make”
   2. Выполнить команду “ls -l module.dwarf” для проверки успешного получения структуры ядра OC
10. Создание профиля Volatility для Metasploitable
    1. Подготавлиаем архив с файлами профиля ядра с помощью команды “zip /var/www/UBUNTU-MSF804.zip /var/tmp/volatility-2.3.1/tools/linux/module.dwarf /boot/System.map-2.6.24-16-server” – профиль своего рода разведовательная информация о системе, в которую мы собираемся получить доступ. Здесь обычно хранятся: процессы, история команд, версия ОС
11. Создание дампа памяти с помощью LiME
    1. Перемещаемся в каталог ‘cd /var/tmp/LiME-master/src’ после чего выполняем команду “insmod lime-2.6.24-16-server.ko "path=/var/tmp/LiME-master/src/mem.img format=lime"” – таким образом мы создаем запись рабочей памяти, где будет хранится информация о состоянии системы, содержимое стека и т.д. Позже мы проанализируем ее
12. Просмотр активности в памяти в момент снятия дампа с помощью Volatility
    1. Для подготовки нам понадобится Python 2. Установка происходит похожим образом с п.7. Выполняется команда “./configure” после чего команда: “make && make instal”
    2. Выполняем команду “update-alternatives --install /usr/bin/python python $(which python2.6) 1” для установки ‘python2’ по умолчанию
    3. Дополнительно скачиваем и устанавливаем пакет ‘pycrypto’ и выполняем команды: “python ./setup.py build” и “python ./setup.py install”
    4. Переходим в директорию ‘/var/tmp/volatility-2.3.1/’ и создаем каталог для архива памяти: “mkdir ./volatility/profiles/”
    5. Выполняем команду “cp /var/www/LinuxUBUNTU-MSF804.zip /var/tmp/volatility-2.3.1/volatility/profiles/” для копирования архива
    6. Выполняем анализ дампа памяти с помощью команды “python ./vol.py --plugins=/var/tmp/volatility-2.3.1/volatility/profiles/ --profile=LinuxUBUNTU-MSF804.zip linux\_lsof -f /var/tmp/LiME-master/src/mem.img | tail -20” – таким образом мы получили расшифрованные данные о стеке действии пользователя и последних обновлениях в памяти