**Студент: Ганиев Р.А.**

**Группа: 11-902**

**Тема: Lime и Volatility. Форензика**

**Содержание**

1. Смена паролей в Metasploitable
2. Обновление источников пакетов
3. Установка LiME
4. Установка вспомогательных библиотек
5. Скачивание, настройка и установка фреймворка Volatility
6. Форензика
7. Результаты работы

**1. Смена паролей в Metasploitable**

Постановка задачи

* смена паролей в Metasploitable
* цель: научиться менять пароль пользователя, сменить пароль пользователя msfadmin
* средства: passwd

Начальная ситуация

* пользователь с именем msfadmin и паролем msfadmin

Выполнение задания

* алгоритм
  + получение привилегий супер-пользователя, команда sudo su -
  + выполнение команды passwd для пользователя msfadmin
  + задание нового пароля для пользователя msfadmin
  + повтор ввода нового пароля
* результаты
  + 
  + пароль пользователя msfadmin успешно обновлён

**2. Обновление источников пакетов**

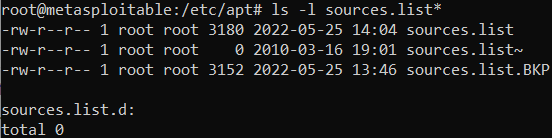
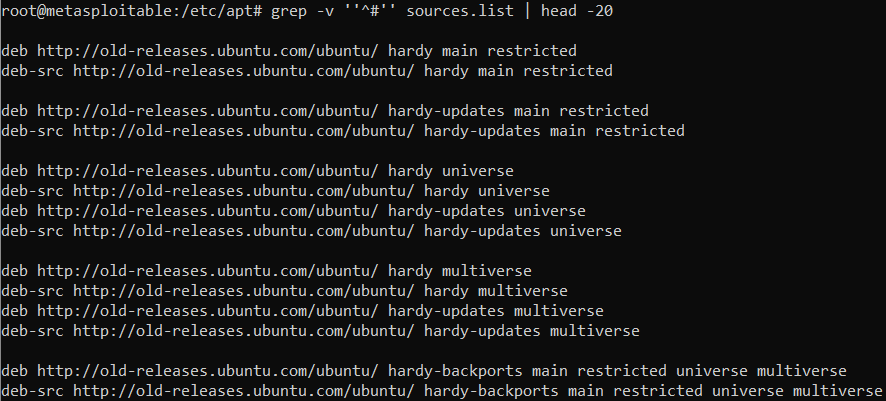
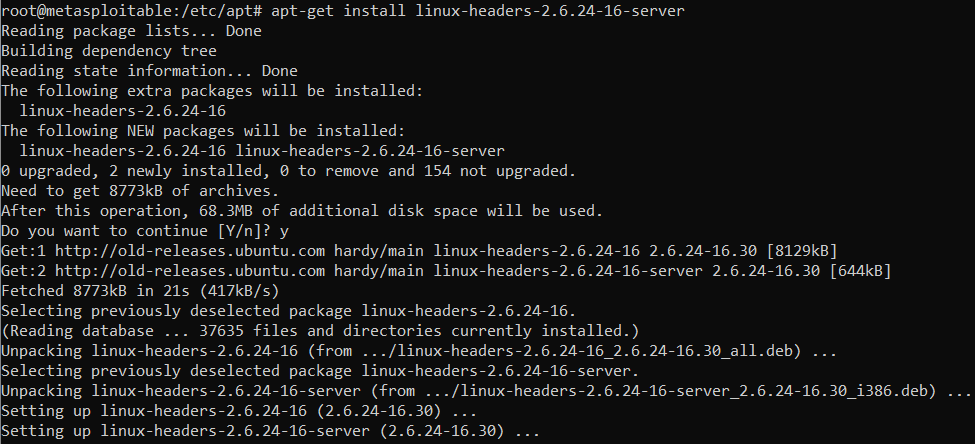
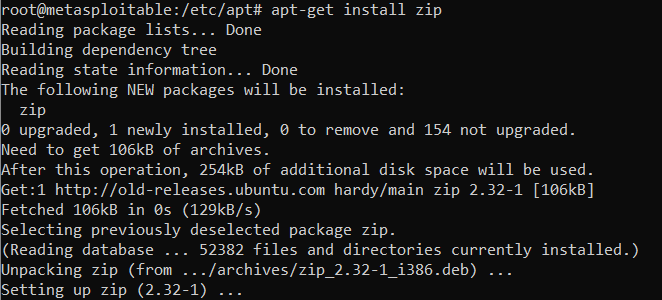
Постановка задачи

* обновление источников пакетов
* цель: научиться менять используемые репозитории, заменить репозитории [http://us.archive.ubuntu.com\*](http://us.archive.ubuntu.com*) на [http://old-releases.ubuntu.com\*](http://old-releases.ubuntu.com*)
* средства: nano, apt-get

Начальная ситуация

* репозитории по умолчанию [http://us.archive.ubuntu.com\*](http://us.archive.ubuntu.com*)

Выполнение задания

* алгоритм
  + создание backup-файла репозиториев
    - переход в /etc/apt
    - создание backup-файл списка репозиториев: cp sources.list sources.list.BKP
    - проверка, что backup-файл создан: ls -l sources.list\*
  + редактирование файла sources.list с целью замены всех репозиториев [http://us.archive.ubuntu.com\*](http://us.archive.ubuntu.com*) на [http://old-releases.ubuntu.com\*](http://old-releases.ubuntu.com*)
  + просмотр списка репозиториев: grep -v ''^#'' sources.list | head -20
  + обновление списка репозиториев для менеджера пакетов apt: apt-get update
  + установка заголовочных файлов ядра
    - просмотр версии ядра: uname -r
    - установка заголовочных файлов ядра: apt-get install linux-headers-2.6.24-16-server
  + установка утилиты Zip: apt-get install zip
* результаты
  + 
  + 
  + 
  + 
  + 
  + 
  + указанные утилиты установлены с репозиториев [http://old-releases.ubuntu.com\*](http://old-releases.ubuntu.com*), а не с заданных по умолчанию репозиториев [http://us.archive.ubuntu.com\*](http://us.archive.ubuntu.com*)

**3. Установка LiME**

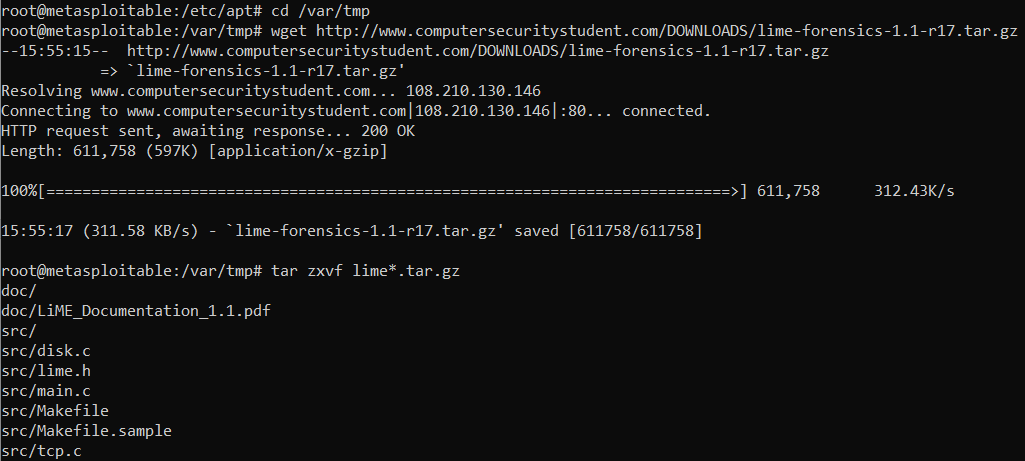
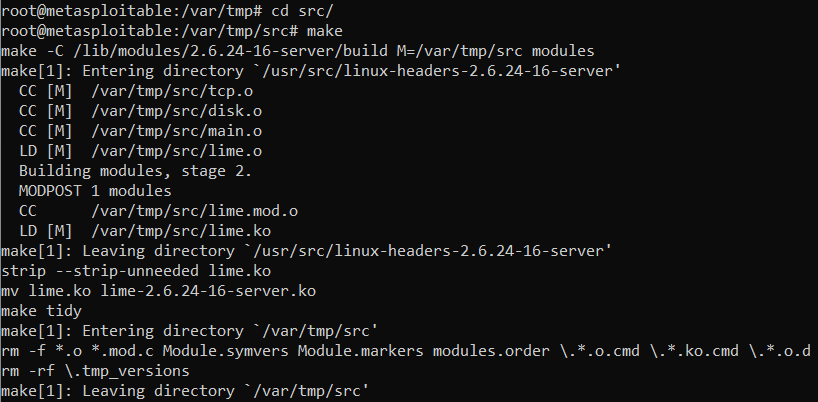
Постановка задачи

* установка LiME
* цель: научиться устанавливать утилиту из архива с исходными кодами, установить LiME
* средства: wget, tar, make

Начальная ситуация

* LiME не установлен

Выполнение задания

* алгоритм
  + переход в /var/tmp
  + скачивание архива с исходными кодами LiME: wget <http://www.computersecuritystudent.com/DOWNLOADS/lime-forensics-1.1-r17.tar.gz>
  + распаковка архива: tar zxvf lime\*.tar.gz
  + переход в каталог исходных текстов: cd src/
  + выполнение сборки: make
* трудности
  + необходимость поиска альтернативных ссылок на архив с исходными кодами
* результаты
  + 
  + 
  + LiME успешно установлен

**4. Установка вспомогательных библиотек**

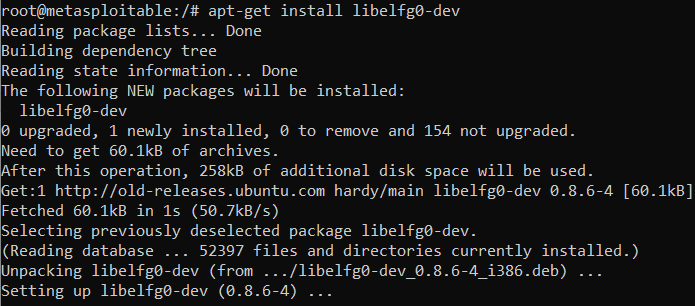
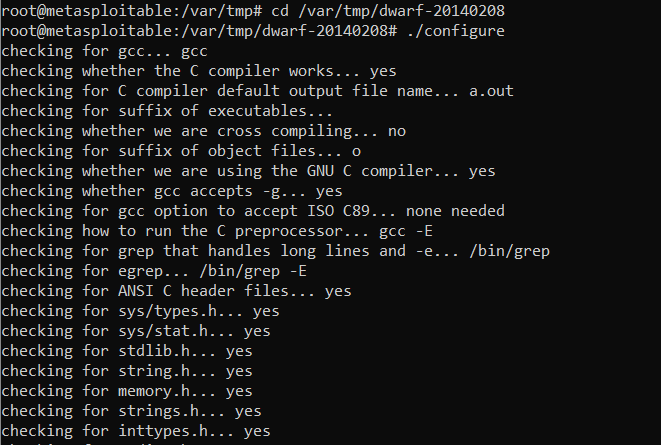
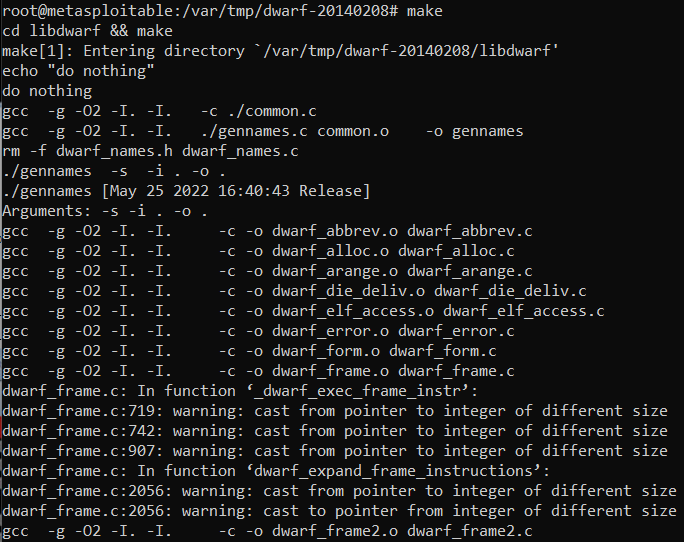
Постановка задачи

* установка вспомогательных библиотек
* цель: научиться устанавливать библиотеки, установить библиотеки
* средства: apt-get, wget, tar, make

Начальная ситуация

* необходимые вспомогательные библиотеки не установлены

Выполнение задания

* алгоритм
  + установка заголовочных файлов библиотеки libelf: apt-get install libelfg0-dev
  + скачивание, настройка и установка библиотеки libdwarf
    - переход в /var/tmp
    - скачивание архива wget http://www.computersecuritystudent.com/SECURITY\_TOOLS/METASPLOITABLE/EXPLOIT/lesson1/libdwarf-20140208.tar.gz
    - распаковка архива: tar zxvf libdwarf-20140208.tar.gz
    - переход в каталог с исходными текстами: cd /var/tmp/dwarf-20140208
    - подготовка исходных текстов к сборке: ./configure
    - выполнение сборки: make
    - перемещение утилиты dwarfdump в каталог исполняемых файлов: cp dwarfdump/dwarfdump /usr/bin
    - проверка правильности переноса утилиты dwarfdump: which dwarfdump
* трудности
  + необходимость поиска альтернативных ссылок на архив с исходными кодами
* результаты
  + 
  + 
  + 
  + 
  + 
  + вспомогательные библиотеки успешно установлены

**5. Скачивание, настройка и установка фреймворка Volatility**

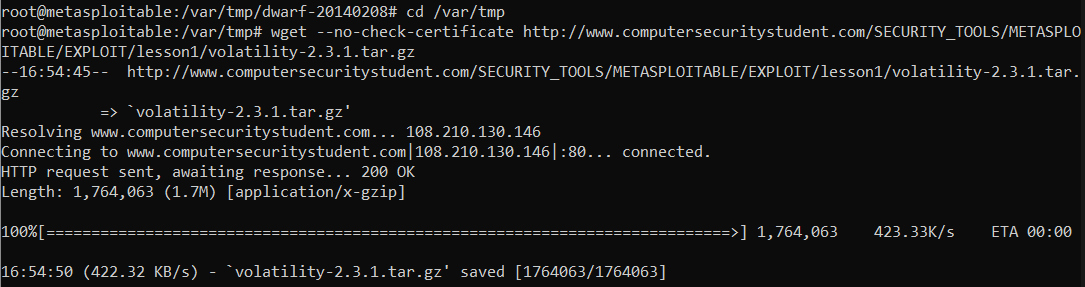
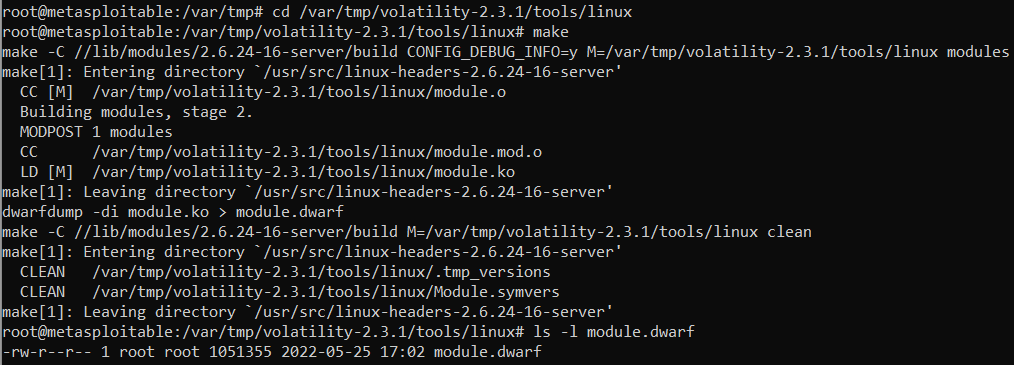
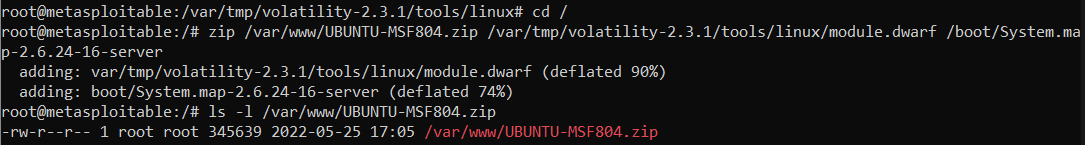
Постановка задачи

* скачивание, настройка и установка фреймворка Volatility
* цель: научиться скачивать, настраивать и устанавливать фреймворк Volatility, установить фреймворк Volatility
* средства: wget, tar, make, zip

Начальная ситуация

* фреймворк Volatility не установлен

Выполнение задания

* алгоритм
  + скачивание Volatility
    - переход в /var/tmp
    - скачивание архива: wget --no-check-certificate http://www.computersecuritystudent.com/SECURITY\_TOOLS/METASPLOITABLE/EXPLOIT/lesson1/volatility-2.3.1.tar.gz
    - распаковка архива: tar zxvf volatility-2.3.1.tar.gz
  + создание файла с отладочной информацией модуля ядра Metasploitable – module.dwarf
    - переход в каталог с linux-утилитами фреймворка: cd /var/tmp/volatility-2.3.1/tools/linux
    - выполнение сборки: make
    - проверка успешности получения структуры ядра Metasploitable: ls -l module.dwarf
  + создание профиля Volatility для Metasploitable
    - выполнение перехода: cd /
    - подготовка архива с файлами профиля ядра Metasploitable: zip /var/www/UBUNTU-MSF804.zip /var/tmp/volatility-2.3.1/tools/linux/module.dwarf /boot/System.map-2.6.24-16-server
    - проверка успешности создания архива: ls -l /var/www/UBUNTU-MSF804.zip
* трудности
  + необходимость поиска альтернативных ссылок на архив с исходными кодами
* результаты
  + 
  + 
  + 
  + 
  + фреймворк Volatility успешно установлен, профиль Volatility для Metasploitable создан

**6. Форензика**

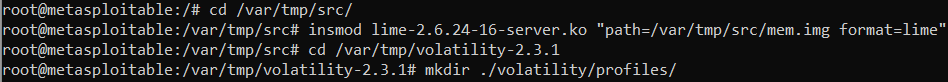
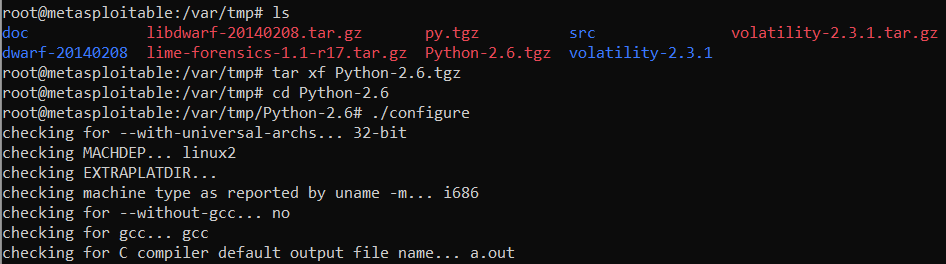
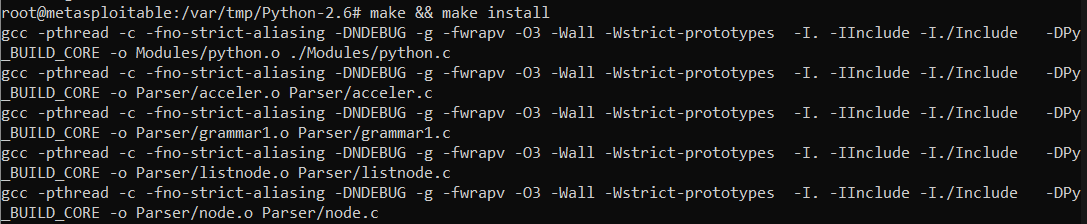
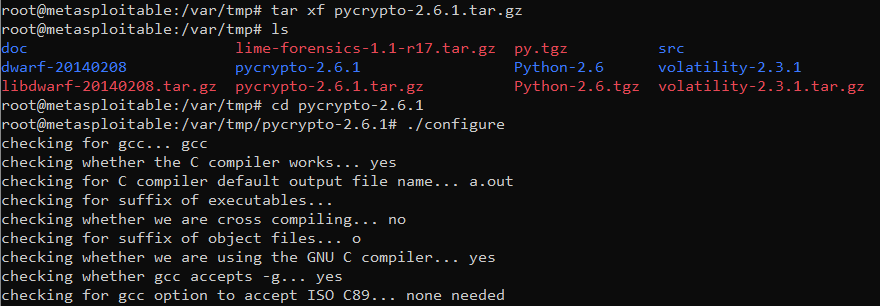
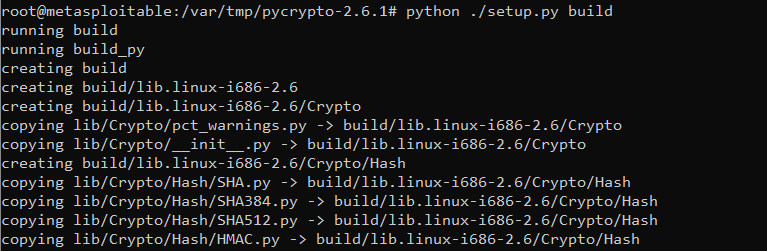
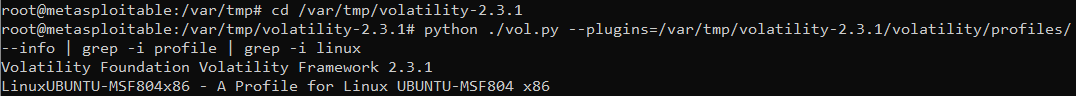
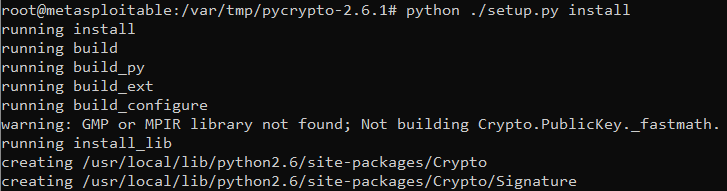
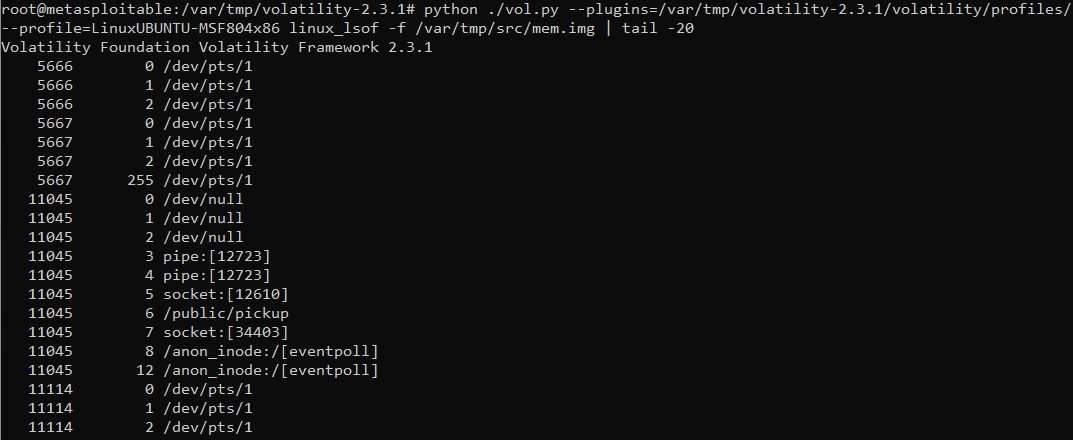
Постановка задачи

* создание дампа памяти и его анализ
* цель: научиться создавать дамп памяти с помощью LiME и просматривать активность в памяти в момент снятия дампа с помощью Volatility
* средства: insmod,

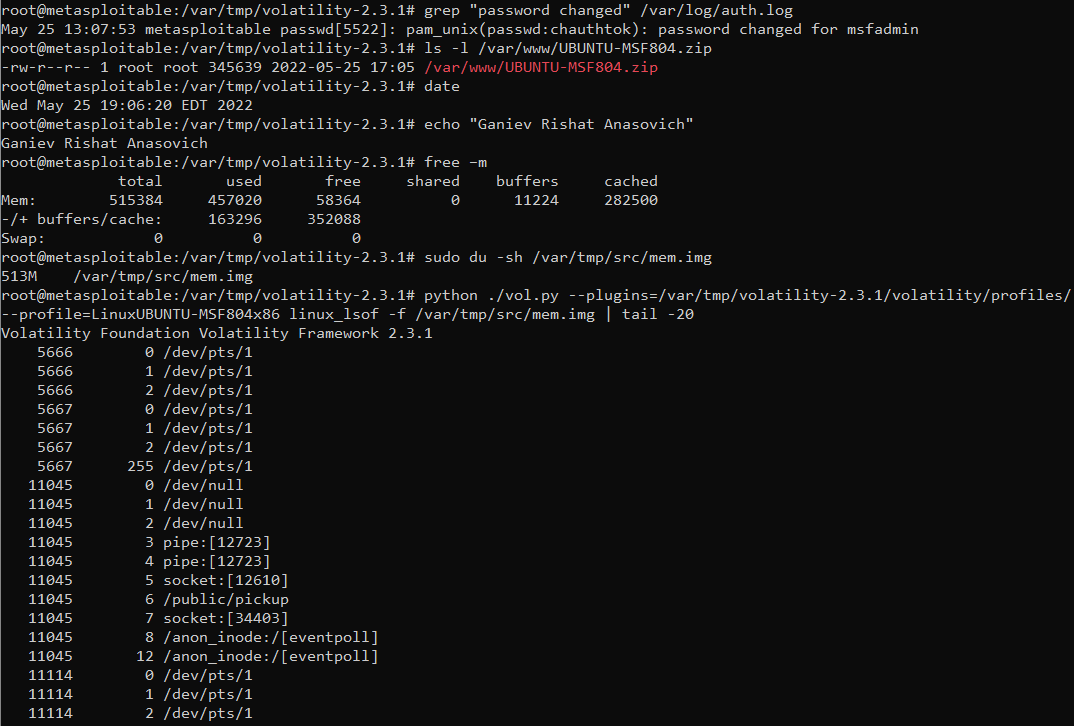
Начальная ситуация

* дамп памяти не создан

Выполнение задания

* алгоритм
  + создание дампа памяти с помощью LiME
    - переход в каталог с собранным для ядра Metasploitable модулем LiME: cd /var/tmp/src/
    - создание дампа оперативной памяти Metasploitable: insmod lime-2.6.24-16-server.ko "path=/var/tmp/src/mem.img format=lime"
  + просмотр активности в памяти в момент снятия дампа с помощью Volatility
    - переход в каталог с фреймворком: cd /var/tmp/volatility-2.3.1
    - создание отдельного каталога для архива UBUNTU-MSF804.zip: mkdir ./volatility/profiles/
    - перемещение архива UBUNTU-MSF804.zip в подготовленный каталог: cp /var/www/UBUNTU-MSF804.zip /var/tmp/volatility-2.3.1/volatility/profiles/
    - установка Python 2.6
    - установка PyCrypto
    - проверка того, что volatility успешно распознала новую директорию с профилями: python ./vol.py --plugins=/var/tmp/volatility-2.3.1/volatility/profiles/ --info | grep -i profile | grep -i linux
    - выполнение анализа дампа памяти (файл mem.img) путем запуска плагина linux\_lsof из состава фреймворка volatility: python ./vol.py --plugins=/var/tmp/volatility-2.3.1/volatility/profiles/ --profile=LinuxUBUNTU-MSF804x86 linux\_lsof -f /var/tmp/src/mem.img | tail -20
* трудности
  + необходимость обновления версии Python с 2.5.2 до 2.6
* результаты
  + 
  + 
  + 
  + 
  + 
  + 
  + 
  + 
  + 
  + 
  + 
  + дамп памяти успешно создан и проанализирован

**7. Результаты работы**

****