"Hacking w made by Macking w

Kacper Ostrowski 2D TŁ

Cel prezentacji

- Przedstawienie:
 - Kilku praktycznych projektów
- Zastosowania wiedzy w praktyce Podatności w sieciach
 - Inżynierii wstecznej
 - Oraz innych praktycznych aspektów hackingu

Przed rozpoczęciem

- Jak dzielimy hackerów ?
 - Black hats hackerzy którzy wykorzystajują swoją wiedze do nielegalnych działalności
 - White hats hackerzy którzy wykorzystajują swoją wiedze do pogłębiania swoich umiejętności oraz do rozwiąaywania problemów, nie wykorzystują tego do żadej nielegalnej działaności
- Kto to jest hacker?
 - (Ang.) A computer hacker is any skilled computer expert who uses their technical knowledge to overcome a problem.
 - (Pol.) Haker komputerowy to każdy wykwalifikowany ekspert komputerowy, który wykorzystuje swoją wiedzę techniczną do rozwiązania problemu.
 - Źródło: https://en.wikipedia.org/wiki/Hacker

Jak dzielimy Hacking?

- Hardware hacking
 - Zmiana, modyfikowanie lub ulepszanie sprzętu oraz jego wewnęrznej logiki
- Software hacking
 - Zmiana, modifikowanie, ulepszanie lub inżynieria wsteczna oprogramowania
 - Network hacking
 - Badanie, exploitacja, przechwytywanie lub wykrywanie podatności w sieciach

Przedstawione projekty

- License_key_crack (software hacking)
 - Złamanie klucza w prostej aplikacji, weryfikującej licencje
- Wave_files_hacking (software/hardware_hacking)
 - Generowanie oraz modyfikowanie plików .wav
- JPG_capture (network hacking)
 - Wychwycenie danych w niezabezpieczonej sieci komputerowej
- Portable lab (hardware/software hacking)
 - Wykorzystanie mikrokomputera raspberry pi zero W do stworzenia przenośnego routera z wbudowanym access pointem do trenowania umiejętności hackerskich

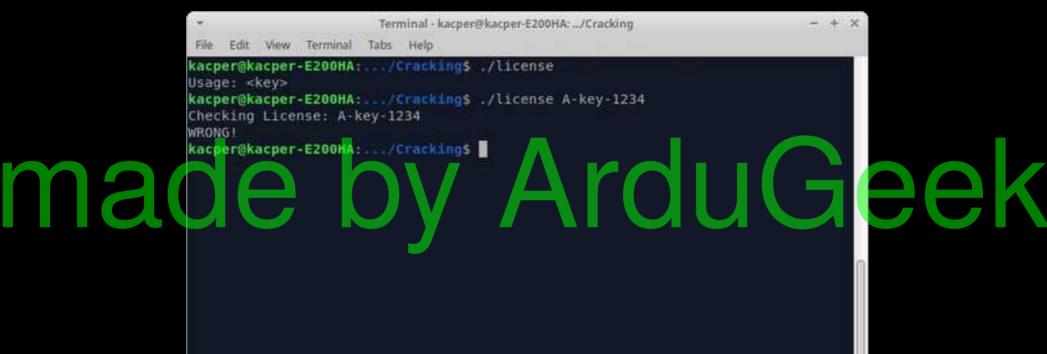
License key crack made by Arducek

Środowisko oraz narzędzia

- Kompilator GCC
- Powłoka systemowa Xubuntu 18.04 LTS (bash)

mebuggerGDBy ArduGeek

Złamanie prostej skompilowanej aplikacji wymagającej klucza do uruchomienia



Kod aplikacji

```
#include <string.h>
      #include <stdio.h>
    pint main(int argc, char *argv[]) {
              if(argc==2) {
    printf("Checking License: %s\n", argv[1])
              if(strcmp(argv[1], AAAA-Z10N-4
    曲
                  printf("Access Granted!\n
                else {
                  printf("WRONG!\n");
10
1616
12
          } else {
    13
              printf("Usage: <key>\n");
14
15
          return 0;
16
```

```
Terminal - kacpen®k
     Edit View Terminal Tabs Helo
(qdb) disassemble main
Dump of assembler code for function main:
   6x888888888886da <+6>:
                                         %rbp
                                  push
                                         %rsp,%rbp
   0x900000000000000db <+1>:
                                  mov.
  8x98888888888886de <+4>:
                                         58x18,%rsp
                                  sub
   8x800800000000006e2 <+8>:
                                         %edi, -0x4(%rbp)
                                  mov
   0x00000000000000665 <+11>:
                                         "rsi, 0x10( rbp)
                                  BOV
   8x88888888888888888 <+15>:
                                         $8x2.-8x4(%rbp)
                                  cmpl
   8x8888888888888888 <+19>:
                                         0x748 <main+110>
                                  ine
                                         -0x10(%rbp).%rax
   0x000000000000006ef <+21>:
                                  may
   0x000000000000006f3 <+25>:
                                  add
                                         S0x8.%rax
   0x000000000000006f7 <+29>:
                                  mov
                                         (%rax),%rax
   8x88888888888886fa <+32>:
                                  mov
                                         arax.arsi
                                         exec(trip). trdi
   8x88888888888888fd <+35>:
                                  Lea
                                                                  # 0x7e4
                                         sexe. leax
   8x9999999999999784 <+42>:
                                  MOV
   8x88888888888888789 <+47>:
                                  callg
                                         8x5a0 <printf@plt>
                                         -0x10(%rbp),%rax
   8x8888888888888888888 <+52>
                                  mov
         000000000712
                                         30x8, urax
              0000
                                          rax).
                         63>:
                                                                    0x7fa
                                           xda( rip), rsi
              00000
                         70>:
                                         Frax, Wrd
                                         8x5b8 <strcmpdplt>
          esee
   0x000000000000000728 <+78>
                                         %eax, %eax
                                  test
                                         8x73a <main+96>
   8x986866666666672a <+86>
                                  ine
                               or q <return> to quit---
   Type <return> to continue.
  8x808898888888872c <+82>:
                                  Lea
                                         0xd7(%rip),%rdi
                                                                  # 0x80a
                                         0x590 <puts@plt>
   8x000000000000000733 <+89>:
                                  callo
                                         0x754 <main+122>
   8x000000000000000738 <+94>:
                                  imp
                                         0xd9(\rip), rdi
   0x0000000000000073a <+96>:
                                  lea
                                                                  # 0x81a
   0x000000000000000741 <+103>:
                                  callu
                                         0x590 <puts@plt>
   0x90000000000000746 <+108>:
                                  imp
                                         0x754 <main+122>
   0x988888888888748 <+110>:
                                         0xd2(%rip),%rdi
                                  lea
                                                                  # 0x821
                                  calla
                                         0x590 <puts@plt>
   8x9888888888888874f <+117>:
  0x800000000000000754 <+122>:
                                  mov
                                         $8x8, %eax
   0x0000000000000759 <+127>:
                                  Leaveg
   8x0000000000000075a <+128>:
                                  retq
End of assembler dump.
(gdb)
```

Deasemblacja kodu aplikacji



```
Terminal - kacpen®kac
    Edit View Terminal Tabs Help
(odb) disassemble main
Dump of assembler code for function main:
   6x888888888886da <+8>:
                                  push
                                         rbp
   0x900000000000000db <+1>:
                                  mov.
                                         rbp, rsp
   8x88888888888886de <+4>:
                                         rsp.0x10
                                  sub
                                         DWORD PTR [rbp-9x4],edi
   8x809899999996662 <+8>:
                                 mov
   0x00000000000000665 <+11>:
                                 mov
                                         OWORD PTR [rbp-0x10], rsi
                                 CMD
                                         DWORD PTR [rbp-8x41.8x2
   8x8888888888888888 <+15>:
   8x80000000000006ed <+19>
                                         0x748 <main+110>
                                 ine
                                         rax.OWORD PTR [rbp-0x10]
   0x000000000000006ef <+21>:
                                  mov
   0x0000000000000005f3 <+25>:
                                         rax.9x8
                                  add
   0x000000000000000f7 <+29>
                                  BOV
                                         rax, OWORD PTR [rax]
   8x8888888888886fa <+32>
                                  mov
                                         rsl.rax
                                         rdi.[rip+8xe8]
   8x88888888888888fd <+35>:
                                  Lea
                                                                # 8x7e4
   8x88888888888888784 <+42>
   8x888888888888888789
                                 call
                                         0x5a0 <printfmplt>
   8x8888888888888888888 <+52>
                                         rax OWORD PTR [rbp-8x 8]
                                  mov
   6x0000000000000712
                                         rax, 8x8
                                                   PTR
          000000000
                                          ax.OWOR
    0000
          00000000
                                          si.[rip+0xda]
                         63>:
                                                                  9x7fa
          0000000007
                        470> :
                                         6x5b8 GTECTED ATES
    x8988888888887
   0x000000000000000728 <+78>:
                                         8x73a <main+96>
   8x986866666666672a <+88>
  Type <return> to continue, or q <return> to quit---
   8x0000000000000072c <+82>:
                                  lea
                                         rdi,[rip+8xd7]
                                                                 # 0x80a
                                         0x590 <putsoplt>
   0x000000000000000733 <+89>:
                                  call
                                         0x754 <main+122>
   8x000000000000000738 <+94>:
                                  imp
                                         rdi,[rip+0xd9]
   0x0000000000000073a <+96>:
                                  lea
                                                                # 0x81a
  0x0000000000000000741 <+103>:
                                 call
                                         0x590 <puts@plt>
   0x90000000000000746 <+108>:
                                 imp
                                         0x754 <main+122>
                                         rdi,[rip+0xd2]
   0x9808080808888748 <+110>:
                                  lea
                                                                # 0x821
                                 call
                                         0x590 <puts@plt>
   8x9898989898989874f <+117>:
  0x80000000000000754 <+122>:
                                         eax, 0x0
                                  mov
   0x0000000000000759 <+127>:
                                  leave
   8x000000000000075a <+128>:
                                 ret
End of assembler dump.
(gdb)
```

Analiza assemblera

user@host:~\$ man strcmp

```
erminal - kacper@kacper-F2
                                                Cracking
     Edit View Terminal
                 (гисар
                         compare
SYNOPSIS
       #include <string.h>
       int stremp(const char *51, const char *52);
       int strncmp(const char *sl, const char *s2, size t n);
DESCRIPTION
       The strcmp() function compares the two strings al and s2. It returns
       an integer less than, equal to, or greater than zero if 31 is found,
       respectively, to be less than, to match, or be greater than 52.
       The strocmp() function is similar, except it compares only the first
       (at most) n bytes of s1 and s2.
RETURN VALUE
       The strcmp() and strmcmp() functions return an integer less than, equal
       to, or greater than zero if $1 (or the first h bytes thereof) is found,
Manual page strcmp(3) line 1 (press h for help or q to quit)
```

Ustawienie breakpointu na początku funkcji main

```
Terminal - kacper@kacper-E200HA: .../Cracking
              Terminal
                      Tabs
   8x8000000000000720 <+70>:
                                         rdi, rax
                                  mov
                                         0x5b0 <strcmp@plt>
   0x00000000000000723 <+73>:
                                  call
   0x00000000000000728 <+78>:
                                  test
                                         eax, eax
   0x0000000000000072a <+80>:
                                         0x73a <main+96>
                                  ine
  -Type <return> to continue, or a <return> to guit---
                                         rdi,[rip+0xd7]
   0x0000000000000072c <+82>:
                                  lea
                                                                 # 0x80a
   0x00000000000000733 <+89>:
                                  call
                                         0x590 <puts@plt>
   0x00000000000000738 <+94>:
                                         0x754 <main+122>
                                  1mp
     00000000000073a <+96>:
                                  lea
                                         rdi frip+0xd91
   0x00000000000000741 <+103>:/
                                  call
                                         0x590 <puts@plt>
   0x0000000000000746 <+108>:
                                         0x754 <main+122>
                                  imp
   0x0000000000000748 <+110>
                                  lea
                                         rdi rip+0xd21
                                  call
                                         0x590 <puts@plt>
   0x0000000000000074f <+117>:
                                         eax.0x0
   0x00000000000000754 <+122>:
                                  mov
                                  leave
   0x00000000000000759 <+127>:
   0x000000000000075a <+128>:
                                  ret
End of assembler dump.
(qdb) break *main
Breakpoint 1 at 0x6da
(adb) run
Starting program: /mnt/mmc-SD64G 0xdabeab8b/HACKING/Cracking/license
Breakpoint 1, 0x00005555555546da in main ()
(qdb)
```

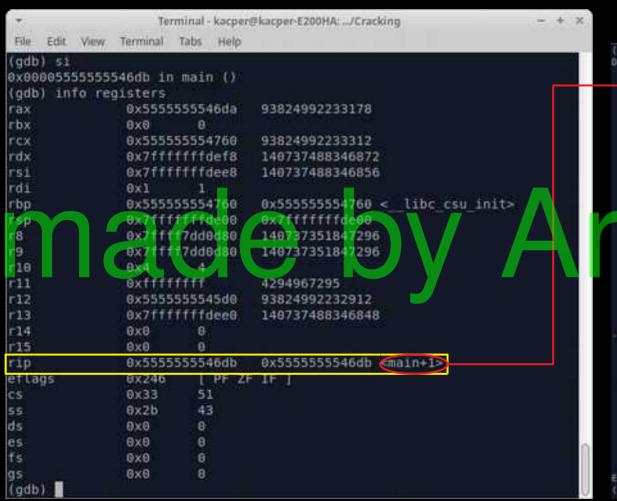
ma

```
Terminal - kacper@kacp
 File Edit View Terminal Tabs Helo
                                  call
                                          0x590 <puts@plt>
   0x0000000000000074f <+117>:
                                          eax. 0x0
   8x8888888888888754 <+122>:
                                   mov
   8x8898888888888759 <+127>:
                                   Leave.
   0x0000000000000075a <+128>:
                                   ret
End of assembler dump
(gdb) break *main
Breakpoint 1 at 0x6da
(adb) run
Starting program: /mnt/mmc-SD64G 0xdabeab8b/HACKING/Cracking/license
Breakpoint 1, 0x00005555555546da in main ()
(gdb) info registers
                8x5555555546da
                                  93824992233178
гах
rbx
                8x8
                8x555555554768
                                  93824992233312
rdx
                0x7fffffffdef8
                                  140737488346872
                8x7fffffffdeet
                                  140737488346856
rs1
(This
                         5/5/178
                                    555555554760
                        ffdeð
                                    x7ffffffffde08
rsp
                        ddeda
                                   46737351847296
г8
F9
                                    48737351847296
r10
                0 X 4
r11
                                  4294967295
r12
                0x5555555545d8
                                  93824992232912
r13
                0x7fffffffdee8
                                  140737488346848
r14
                8x8
r35
                                  0x55555555546da <main>
гір
                0x5555555546da
                          [ PF ZF IF ]
eflags
                0x246
                0x33
                         43
                8x2b
ds
                8x8
                          8
                0x0
es
                ByB
--- Type <return> to continue, or g <return> to guit---
gs:
                0x0
(gdb)
```

Sprawdzenie wartości rejestrów

```
Dump of assembler code for for the side
   0x000000000000000da <+0>:
                                 push rbp
   0x800080000006db <=1>:
                                        rtip, rsp
   0x8000800000006de <+4>:
                                        rsp.0x10
                                        DWORD PTH [rbp-0x4]_edi
   0x8000000000000662 <*8>:
   0x000000000000006e5 <+11>:
                                        OWORD PTM [rbp-0x10] rs1
                                        DWORD PTR [rbs 8x41,0x2
   8x800000000000006#9 <*13>
   0x800000000000006ed <+19>
                                        me748 -cmain+118>
   0x8800800000066f <+21>
                                        rax, OMORD PIR [rbp-ex10]
   0x00000000000006f3 <+25>
                                         Prop. 0 v B
   0x80000000000000
                  5f7 <+29>
                                        CDX, ONGRO PTR [rax]
                       +35>
                                         rdi Frip+0x
           000000
                   784
                        +42>
           0000000
                   100
                       457>
                                                   Table Street
                                               ORD PTR I TOP OX
   0x8000800000000716 <+60>
                                        raw_OWORD FIR [rax]
   0x000000000000000719
                                 lea.
                                        rsi,[rip+0xda]
                                                               # 0x701
   0x00000000000000726 <**70>
                                        0x5b0 <strempDelt>
   0x80000000000000723 <+73>
                                 call
   0x80000000000000728 <+78>
   0x90000000000000072m <*88>:
                                        Dx73a <main+96>
   Type <return> to continue.
                               Of a sceture to ourte
   0x0000000000000072c <+82>
                                        rdi. [rin+0xd7]
                                                               # DXDDs
   0x00000000000000733 <+89>1
                                        0x590 <putsoplt>
   0x8000000000000738 <+94>:
                                        0x754 <main+122>
   0x800000000000073a <+96>:
                                        rdi.[rip+0xd9]
                                                               # BXELD
                                        0x590 <puts0plt>
   0x80000000000000741 <+103>:
   9x0000000000000745 <+108>
                                        0x754 <main+122>
   8x8806889999999748 <+119>
                                 tea
                                         rdi, [rip+0xd2]
                                                               # 0X801
   0x800080800000074f <+117>
                                        8x590 <putsonlt>
   0x80000000000000754 <+122>
                                 nev.
                                        ear, exc
   0x000000000000000759 <+127>:
                                 Leave
   0x000000000000075a <*128>:
                                 TEL
End of assembler dump,
(gdb)
```

Kilka adresów dalej ...



```
(adb) disassemble main
Dump of assembler code for function main:
   0x0000000000000000da **0>:
                                 push
                                        rbs
                                        rtip, rsp
                                        rsp. 0x10
                                 sub
  DxRnonnonnonnonnone2 <*R>:
                                 MOV
                                        DWORD PTH [rbp-0x4].edi
  0x000000000000006e5
                                        OWORD PTM [rbp-0x10], rs1
                                        DWORD PTR | rbs-0x41.0x2
                                 ine
                                        se748 <sain+118>
                                        rax, OwoRD PIR [rbp-ex19]
                                 MOV
                                 add
                                        nax. 0x8
                                        ray, OWORD PTR [rax]
                                        rsi, rax
            0000006frt <+35>
                                 Lea
                                        rdi,[rip+0xe0]
                                                               7 11 7
                       -1.7
                 00709
    квасев
                 9070e
                00712
                        0.110
                                         ON THE
                                                               ( ) ( ) ( ) ( )
                                        0x5b0 <strempdelt>
   0x80000000000000723 <+73>
                                 call
                                 Service -
  0x8600000000000072a <+88>:
                                        0x73a <main+96>
                              or q <returns to omit
   Type <return> to continue.
  rdl.[rip*9xd7]
                                                               # DXDDs
  8x000000000000000713 <+89>
                                 call
                                        0x590 <putsoplt>
                                        0x754 <main+122>
  8x866666666666735 <+94>
                                 1mp
  0x8800800000000073a <+96>:
                                        rdi.[rip+0xd9]
                                 Lea
                                                               # DXELD
  0x0000000000000741 <+103>
                                        0x590 <puts0plt>
  9x80000000000000746 <+108>
                                        0x754 <main+122>
  0x880888888888999748 <+119>
                                 Lea
                                        rdi, Fipemed2+
                                                               # 0X891
  0x80000000000074f <+117>
                                 ent!
                                        0x590 <putsoplt>
  5x8888880000006888754 <+122>
                                        max. axa
  0x00000000000000759 <+127>:
                                 Leave
  0x8000008890000075a <+128>:
                                 761
End of assembler dump
(000)
```

```
Terminal - kacper@kacper-E200HA: __/Cracking
              Terminal
(adb) ni
0x000055555555546de in main ()
(adb)
0x000055555555546e2 in main ()
(adb)
0x000055555555546e5 in main ()
(adb)
0x000055555555546e9 in main ()
(qdb)
0x000055555555546ed in main ()
(adb)
0x00005555555554748 in main ()
(adb)
   00055555555474f
  Ь)
(adb)
0x000055555555554759 in main ()
(qdb)
0x00005555555555475a in main ()
(adb)
  libc start main (main=0x5555555546da <main>, argc=1,
    argv=0x7fffffffdee8, init=<optimized out>, fini=<optimized out>,
    rtld fini=<optimized out>, stack end=0x7ffffffffded8)
    at ../csu/libc-start.c:344
344
         ../csu/libc-start.c: No such file or directory.
```

Przejście przez program bez podanego klucza

uGeek

Terminal - kacper@kacper-E200HA: _/Cra

Przejście przez program z losowym kluczem

```
The program being debugged has been started already.
Start it from the beginning? (v or n) v
Starting program: /mnt/mmc-SD64G 8xdabeab8b/HACKING/Cracking/License AAAA-KEY-1234
Breakpoint 1, 0x0000555555546da in main ()
0x000055555555546db in main ()
0x000055555555546de in main ()
0x0000555555555546e2 in main ()
exeee055555555546e5 in main ()
0x000055555555546e9 in main ()
                    10 0410
0x86885555555546ef in
                        18 1 H
0x000055555555546f7 in main ()
0x000055555555546fa in main ()
0x000055555555546fd in main ()
0x00005555555554704 in main ()
0x00005555555554709 in main ()
Checking License: AAAA-KEY-1234
```

Edit View Terminal Tabs Help

(adb) run AAAA-KEY-1234

344

(adb) ni

(adb)

(adb)

(adb)

(gdb)

(gdb

(gdb

(gdb)

(adb)

(adb)

(adb)

(qdb)

(adb)

(qdb)

0x0000555555555470e in main ()

0x00005555555554712 in main ()

../csu/libc-start.c: No such file or directory.

```
Terminal - karpert
File Edit View Terminal Tabs Help
0x00005555555554709 in main ()
(dbp)
Checking License: AAAA-KEY-1234
0x0000555555555470e in main ()
(adb)
0x00005555555554712 in main ()
(ddb)
0x80005555555554716 in main ()
(dbp)
0x000005555555554719 in main ()
(adb)
0x909055555555554720 in main ()
(odb)
0x800005555555554723 in main ()
(gdb)
0x908055555554728 in main ()
         55555472a in muin ()
 adb)
 x808651
         555555473a
(gdb)
WIRLDING I
0x00005555555554746 in main ()
(ddb)
0x80005555555554754 in main ()
(ddb)
0x8000055555555554759 in main ()
(ddb)
0x800005555555555475a in main ()
(adb)
  libc start main (main=0x5555555546da <main>, argc=2,
    argv=0x7fffffffdec8, init=coptimized out>, fini=coptimized out>,
    rtld fini=coptimized out>, stack end=0x7fffffffdeb8)
    at .../csu/libc-start.c:344
         ../csu/libc-start.c: No such file or directory.
(adb)
GVDGDGT####TWDSHDG
```

```
Terminal Tabs
Checking License: AAAA-KEY-1234
0x0000555555555470e in main ()
(adb)
0x00005555555554712 in main ()
(qdb)
0x00005555555554716 in main ()
(adb)
0x00005555555554719 in main ()
(qdb)
0x00005555555554720 in main ()
(adb)
0x00005555555554723 in main ()
(odb) set seax = 0
 alab
 odb
     005555555554728 in main ()
(dbp)
      set seax = 0
(qdb)
0x0000555555555472a in main ()
(sdb) set seax = 0
 (adb) ni
0x0000555555555472c in main ()
0x0000555555554733 in main (
(gdb)
```

Terminal - kacper@kacper-E200HA: __/Cracking

Złamanie klucza

```
(gdb) disassemble main
Dump of assembler code for function main:
   0x0000000000000000da <+0>:
                                  nush
                                         rbp
   rtip, rsp.
   0v80088080000006de <+4>:
                                         rsp. 0x10
   byRnonnonnonnonoccel <*R>:
                                         DWORD PTH [rbp-0x4] edi
   020000000000000000665
                                         OWORD PTH [rbp:0x10], rs1
                                        DWORD PTR [fbp-8x41,0x2
                                  CBD
                                         me748 <main+118>
                                 toov
                                         rax, Oword PTR [rbp.ex10]
                                  add
                                         race. dock
                                         rax, QWORD PTR [rax]
                                  mov
                                  mov
                                         rsi, rax
                                        rdi.[rip+0xe0]
                                                                # 0x7e4
                                  Lea:
                                 \leftarrow111
   0x000000000
                                           all corintfile; to
                                 DOV
                                         CHAIR .. OWICH
                                  add
                    100
                                  mov
                                  lea
                                        0x5b0 estremplist
                                         0x73a <main+96>
  -Type <return> to continue,
                                         rd1.[rip+9x07]
                                                                # DXDDs
                                  tea
                                  call
                                         0x590 <putsoolt>
                                         0x754 <main+122>
                                                                # DYSES
   0x800000000000073a <+96>:
                                         rdi.[rip+0xd9]
   0x0000000000000741 <+103>:
                                  call
                                         0x590 <puts0plt>
                                  fmp.
                                         0x754 <main+122>
                                  Lea
                                         rdi, Fip+mxd2
                                                                # 0X801
   0x800080800000074f <+117>:
                                 mail!
                                         0x590 <putsoplt>
   6x8866600006666754 <+122>
                                 mev
                                         ear, exe
                                  Leave
   0x00000000000000759 <+127>:
   0x8000008880000075a <+128>:
End of assembler dump
(dab)
```

```
Terminal - kacper@kac
    Edit View Terminal Tabs Help
(qdb) disassemble main
Dump of assembler code for function main:
   6x8898888888886da <+6>:
                                  push
                                         rbn
                                         rbp, rsp
   exeeeeeeeeeedd <+1>:
                                  mov.
   8x88888888888886de <+4>:
                                  sub
                                         rsp. 0x10
   8x808888888888686e2 <+8>:
                                         DWORD PTR [rbp-0x4],edi
                                  mov
   0x00000000000000665 <+11>:
                                  HOV
                                         QWORD PTR [rbp-0x10], rsi
                                         DWORD PTR [rbp-8x41.8x2
   8x868888888888888 <+15>:
                                  CMD
   0x000000000000006ed <+19>:
                                         0x748 <main+110>
                                         rax.OWORD PTR [rbp-8x18]
   0x000000000000006ef <+21>:
                                  mov
   0x000000000000005f3 <+25>:
                                  add
                                         rax, 9x8
                                         rax, OWORD PTR [rax]
   0x000000000000006f7 <+29>:
                                  BOV
   8x88888888888886fa <+32>:
                                         rsi.rax
                                  mov
                                         rdi.[rip+8xe8]
                                                                 # 8x7e4
   8x88888888888888fd <+35>:
                                  Lea
                                         eax. 0x0
   8x8688686868888784 <+42>:
                                  MOV
   8x88888888888888789 <+47>:
                                  call
                                         0x5a0 <printf@plt>
                                         rax OWORD PTR [rbp-8x18]
   6x0000000000000070c <+52>:
                                  mov
                                  add
    x0000000000000712 <+5
                                         rax, 0x8
                                           x. OWORD
                                                    PTR
           000000007
           100000007
                         100
                                           i.[rip+8xda]
                                                                   0x7fa
     9888888888888
                         70> :
                                          ACTUAL TO A LINE HOLD AND
   0x00000000000000728 <+78>:
                                         eax, eax
                                  test
                                         0x73a ==ain+96>
   8x98686666666672a <+88>:
                                        turn> to guit---
   Type <return> to continue.
                                or q <r
   8x00000000000000072c <+82>:
                                  lea
                                         rdi,[rip+0xd7]
                                                                 # 0x80a
                                         0x590 <puts@plt>
   0x000000000000000733 <+89>:
                                  call
   6x00000000000000738 < 945
                                         0x754 <main+122>
                                  jmp
   0x000000000000073a <+96>
                                         rd1;[r1p+8xd9]
                                                                 # 0x81a
                                 Lea
   0x000000000000000741 <+103>:
   0x90000000000000746 <+108>:
                                        rdi,[rip+0xd2]
                                                                 # 8x821
   0x988888888888748 <+110>:
   8x0666000066000874f ++117
                                         0x590 <puts@plt>
   0x00000000000000754 <+122>
                                         eax,0x0
   6x8888888888888759 <+127>
                                  Leave
   0x000000000000075a <+128>:
End of assembler dump.
(gdb)
```

Co się tak naprawdę stało ?



ek

Wave files hacking made by Atducking K

Środowisko oraz narzędzia

- System Xubuntu 18.04 LTS
- Interpreter Python 3.6

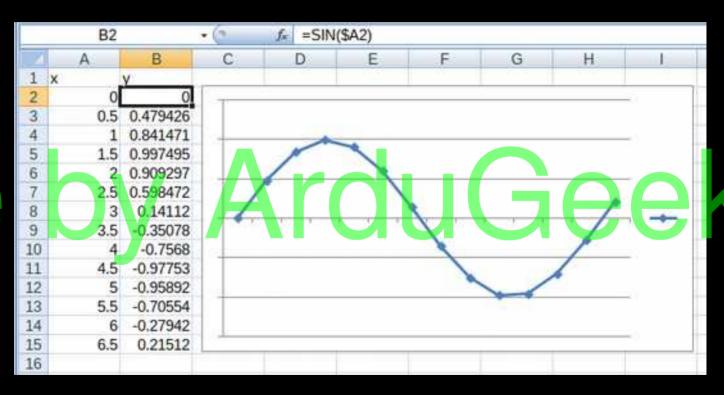
Marduino Dodatkowo: La Carduino La Carduin

- Oscyloskop Cyfrowy
- Filtr dolnoprzepustowy RC

Parametry pliku .wav

- Bit rate
- Bit depth

made



13_D = 1011_{BLE} 13_D = 1101_{BBE}

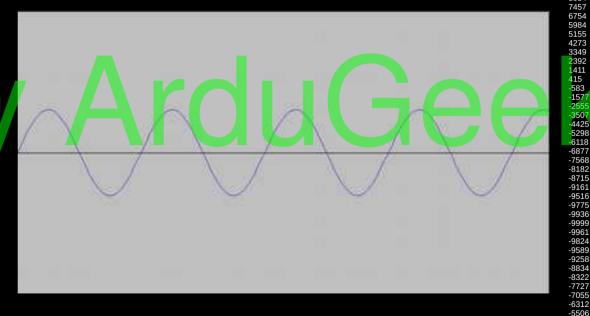
Parametry cd.

```
Terminal - kacper@kacper-E200HA: .../Audio
                       Tabs Help
          View Terminal
                                                                  RIFF.:..WAVEfmt
00000000
                                                          74 20
00000010
00000020
00000030
00000040
00000050
99999969
                           0d 58 09
00000070
00000080
                                      30 d
                                                       19 a0 d9
00000090
kacperaka
           RIFF (little-endian) data, WAVE audio, Microsoft PCM, 16 bit, mono 44
sound.way
100 Hz
kacper@kacper-E200HA: /Audios |
```

Jak wygenerować plik .wav

```
import wave, struct, math, random, numpy
      sampleRate = 44100.0
      duration = 100.0
      frequency = 440.0
      obj = wave.open('sound.wav','w')
      obj.setnchannels(1) # mono
      obj.setsampwidth(2)
      obj.setframerate(sampleRate)
      audio = []
               numpy
                    .arange(0
               values:
                 int(10000*numpy.sin(d)
          math sine.append(var1)
    Efor a in range(10000):
16
          for b in math sine:
              audio.append(b)
    Efor c in audio:
19
          data = struct.pack('<h', c)
20
          obj.writeframesraw( data )
      obj.close()
```

216=65536 65536/2 = 32768 Bit depth = (-32768, 32768)



4794

7173

9320 9635 9854

-157

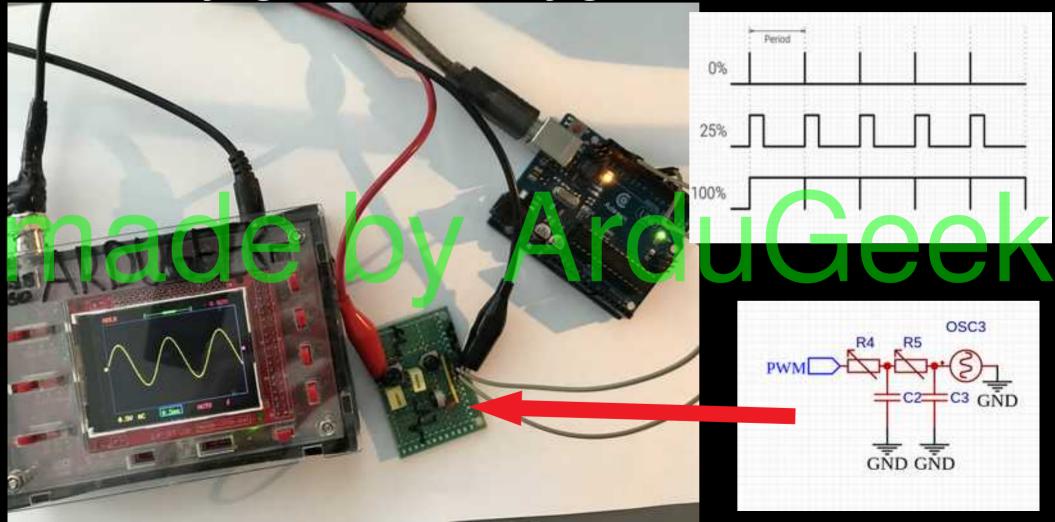
-3738



Do czego to może służyć

- Możemy tą metodą spreparować sygnał cyfrowy:
 - Przy pomocy wzmacniacza I anteny wemitować go zakłucając lub uzyskując porządaną łączność
 - Przy pomocy arduino odtwarzać ten sygnał z karty microSD i wprowadzić go do danego układu uzyskując porządany efekt

Prosty generator sygnałów z arduino

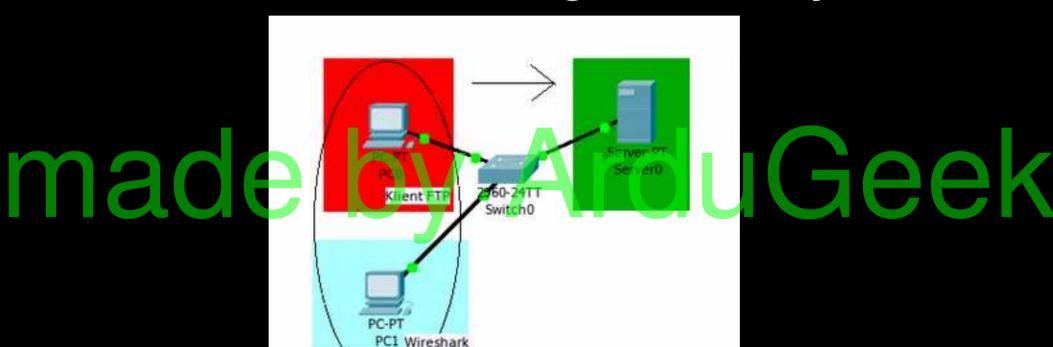


JPG capture made by ArduGeek

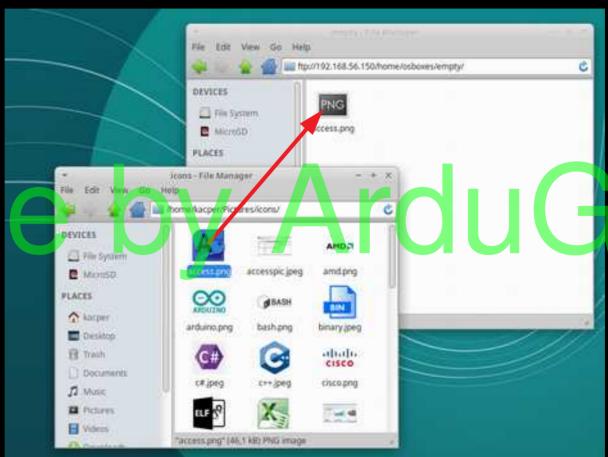
Środowisko oraz narzędzia

- Maszyna wirtualna Ubuntu Server 18.04 LTS z skonfigurowanym serwerem FTP, oraz statycznym adresem IP 192.168.56.150/24
- System Xubuntu 18.04. LTS
 - Oprogramowanie wireshark
 - Menedżer plików Thunar

Przechwycenie pliku przesyłanego w źle/słabo skonfigurowanej sieci

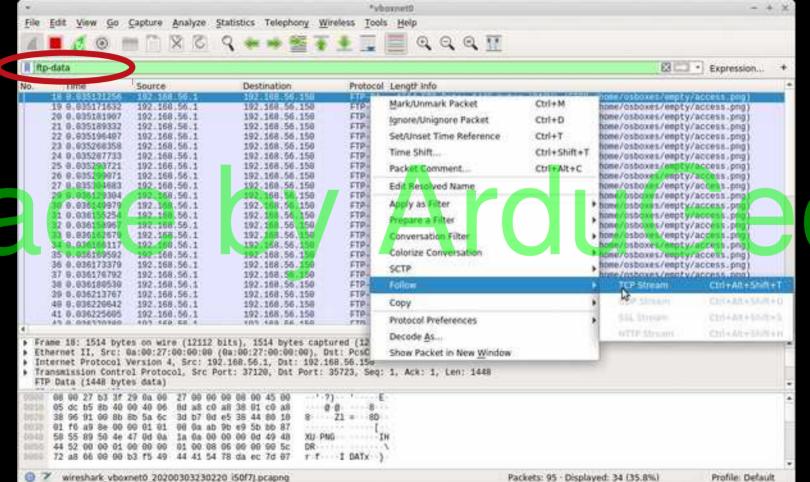


PC0



mad





PC1 cd.

Wireshark - Follow TCP Stream (tcp.stream eq.2) - vboxnet0 005c72a8660000n3f54944415478daec7d07805d65d1f6bc67376537bd77120 209a184d07b68099d58b0018a6a8a821a44f4b3fc82929f140122d58a9f6241d028063420a208a87442e88190109290906c78b3c9d67bef99ffb4f7 hc33f3ceb9bb618346c9d570efde72eeb9e79c9977e695679e31f0f6b8197a7fc7e1df1c3fb8f7R8a9b500db1b84f108669431382c7a6970f4773f0 4a88f1ef78efef592c95a88de193d17fd1fcad1835e183d2c477fc42fc4ff2fc79b458cb75e13fd9d7c26faab267aba267aba267a6b197dc6c4db81 f83ed98de46fc964b3c97f4cf45e93bc92bc97926d265b45fb33b2bd481f19937c3af9eef4dbc98e264f647fa4df91ee63fa7cf291fce860be9bf63 b4dfe5de927e97be93eb8c7ec75fb7d89907f0df98ef43d86fc6d5f7347c3d967307f13390ec6be5hec3fb8edb09f657f15b2cdb8efcc7e60fe3e71 bcd33394ef@b66a730df4bb2cf765fe5368@fd8c64bfc364af21fb1dca6f2107d43efa48c3@ffb467761f68ed86dbcfb86f15f6bf0f71d75d5@e754 led096682fda383b25374544647e7b13e3aee263f98e8cea5f12e9e024389bd81f89b1919f20bda9e61f2505ea1ee73869f6c64db96ef6acab7650d 3edb8641e542e64e243d50e4c26306283e6f84b15883d78e993496c38c471a323900e4bfc94f61fb677fa711c79ed86f34f47cd9cf479ecf84d6bd6 2622ae298737b4564bf813a39e644911bb7fd7edd11992b8c9e3f7a7d297754dcf9785e21769983a2bb4a7c49569ea9d25d2769fedb8cfecf475d31 79494ddf8f1a638e8cRedc2ed193c3a20318f88b921b1f3dbf26390968b99de7d71bbb3222e33749045866db327c55b41719fd2cbd48e26bb8c950c 1fb4e6b9cd4c1b8eda4df1d02bd56e80f538d015e48a81d1f20ab3882b7e0938b98acc77231e64e94ad96e4bb992db9c8c1c828c2708347e9cd50ec 3373dc46ice3f45ceb8e50f97e16i545e74efe6ef2f399d39217198a636b507caefaf1264729d993c8018c891e7466ff4a9903a8bc1dic40b2ff7f3 #3bfbd507bd2e12fddcc3x3#1264797aa8f14ac9u5ab8c8af6b79e65df89c84fd98e661897c69c538a4a6df85d489c848835e2cf4c2b af07f9397edia2cb0b48182e236711cab3eb9a87a1c6bec7187f2544e385ec76554690ce0e599a62bd1e324f077ad4951f4f933b3a6aa9b9711a le71b436650e97b51acc0ce81e6e726fe0dce83a6869ebc47a697329293d11d1247cb531f279df1542977faa1f913e28699b71e3989ffdb31baeb eb5667799908388ff5b1s8069666558386f71e786ef4f8c4e8984c899eed4dc3e7f42262caf32ca859f67d3624545c568ecca4fc34a632f13f 29437d9ab31986c47285f652982d2c26612218b17cocf726799e2f8a7c154ab68b21777e2c27e6171c5by96c3442ec1ad97e83cf8bd986d acc5891adc4f698d9efa4c62f7368796e4989d9fc94ch3b77228299148e48h946289e221gabcc3df3f3ef479b5e5a95dce1448e6 a74d716fd6b214ea0f4dfe689927d7df558eb4eaa1.3567463fff89c5988c7287c414832fd941351a9644bcb1bde4b861a65995be829ac2fc9a2d 587d1729759fd1b65ea2955363fef46c6698b6771958f986fa665ce94aebec9874368b62b4279c37e3bdg9fecd2341a4868f7d3783416b981a4 44-f4b42-40f-64-6b133562-een14-d19f7-nac33fb35-n8-69117431c82fb51a39f27462494-n92668-d8-69-552 48f6f191e36ef85c98e6c3d191d9353a2eb5967450cb479c459e1e1fc8c949af21fda966791df48efed59ce89bb5d1bf9afef55953df6b82bed1bf3 eb510f4ae81a@7ff4b8772d98da99b02689dae870t3dd27ff929369fd2608b23d351886915144c9614d6d4d7781d8ae6f9bb11963buf7f628385be4 fbb6d49f14c1f956b28fddfc52fcd7ecea6dd3ce 151849fadf9f36fce4c9e8c1a5cc916ccaa281ceff7497699dfffcc9980f4112e6c7cb90e17925cdb5f262982d5799a414d667fc19a8df7e18d44d1 a097d278e89ee87499f3183deda9d37263748fb78db6ddbad27b79b879fa138cac4911c123dd818fddb96ddff473b89ccf96ffc42b4487d2cfa8153 5dcd993c90c44354a35ba721f5593f6d14f49f3616fa4d1b99f5534724abf8bff54799f4a76d7398db6e3d7199b20619df4eff2f7296e6d1e3af3fc 549705ef493f78cfe9c345297af25a7ab7fedb97a18b9fb5818b9db94e83f63cc5bbec26f8998619b33d8767b3311894928fe2b2299f3d7a3afde71 48af3e57448f8f8e7e595f67e82643846989c8415293f79a18fd9h8883f71e8f7d278cf98f3cb1db9cc1b65b776e3f1d7e3a4f7bb347a737fcf43fd 60124dfffd809377e3dfs1d9f881e8ef1d114dfe7d50ea983c1fb4d8281976c0f83f6d96ef3bf356678b07fd973b6b485ebe8213d44a8223cbc4495 a1f3b2cc64f9bc5ebddb1e94f47eb2c9df93d34eb2d21df8d93ddf131d89427ffb4965c4238b8157ce734f187f7b949587f4ac8ab69eb7b786fc6e7 a514b8a39df1d654f5895c264687fce9d299f349b81cca5c7c6e4ac41b7e4B87ba9189481fc7c982ecd4c6e17a1d7d4914a9470babc5292bfff531d 80f9dbh1d7ecd9bfa6f755d18f9819fd98805e85a8c0c27dc60c844107ee08830f9904fda68cdc8cb3191fRee858944330f1bfd8d82bba5153R3ce2 f4b748c7d8345be80925d48947b4c6e6845e6c84823b65d49e1da7bdf8bbcdc98db24d96d217be84a9ef67db1a79bac77cc7e2ffb11829f50ed4033 2b724e228bdce2ed1b1bc51952e2cbcbb4d63118e6d2f8b3da65dc43dcbdd0a68bb6ed3b2d67ea8c79829697edca7d41f4db6d593e48189c267b6c6 fbd761aa146000cface76ebf4f5ff790e20caf56ff84a14f27e267a3886f2e68d5c8da2d7fa8c1e1819fd541872d864a8db7e58f7bea1141de9cee8 32 client pkts. Ø servey pkts. Ø turns Stream 2 : Entire conversation (46 kB) Show and save da Find: Find Next

X Close

Filter Out This Stream

Server0

UbuntuServer [Running] - Oracle VM VirtualBax File Machine View Input Devices Help Libuntu 18.04.4 LTS osboxes ttu1 osboxes login: osboxes Password: Last login: Mon Mar 2 21:03:57 UTC 2020 from 192,168,56,1 on pts/0 Relcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-88-generic x86_64) * Documentation: https://help.ubuntu.com https://landscape.canonical.com * Management : * Support: https://ubuntu.com/advantage System Information as of Tue Mar 3 22:08:04 UT ioud: Processes: of /home 0.0% g Users log 8 IP address for enposit Memory usage: Subg Usage 1 packages can be undated. updates are security updates.

mad

osboxes@osboxes:"# _

Wynik

 Przechwycone dane zapisane pod rozszerzeniem .jpg, będą widoczne jako

made



uGeek

Portable lab made by ArduGeek

Środowisko oraz narzędzia

 Karta microSD z raspbianem lite (odmiana Debian GNU/Linux)

• Mikrokomputer raspberry pi Mikrokomputer raspb

Hardware



Jakie funkcje mogą być przydatne?

- Server ssh
- Server FTP
- Server http
 Server mysql

 ArduGeek
 - Aplikacja phpMyAdmin

Podłączenie do raspberry pi



Skonfigurowane usługi

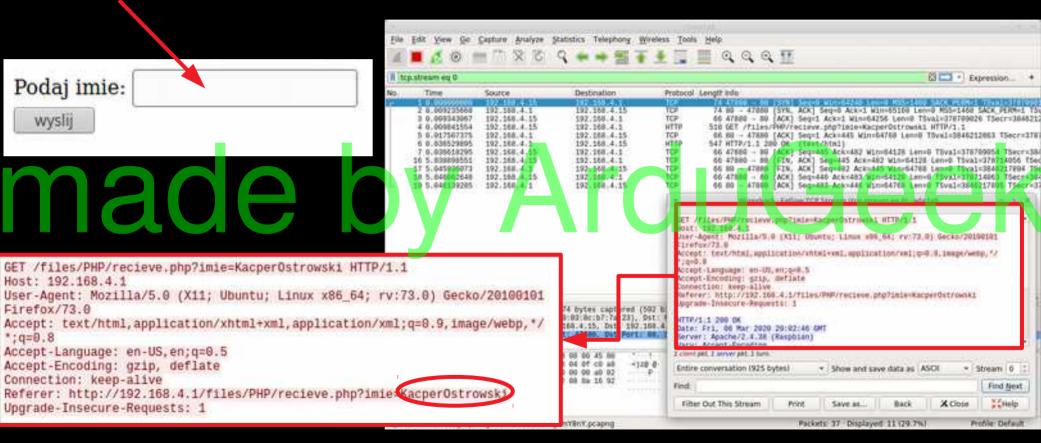
```
kacper@kacper-E200HA: $ nmap 192.168.1.170
Starting Nmap 7.60 ( https://nmap.org ) at 2020-03-06 20:42 CET
Nmap scan report for 192.168.1.170
Host is up (0.0047s latency).
Not shown: 996 closed ports
PORT STATE SERVICE

21/tcp open ftp
21/tcp open ftp
22/tcp open ssh
53/tcp open domain
80/tcp open http
Nmap done: 1 IP address (1 host up) scanned in 19.88 seconds
kacper@kacper-E200HA:-$
```

Do jakich testów może się przydać Portable lab

- Testy aplikacji webowych
- Testy bazy danych
- Przechwytywanie danych przesyłanych w sieci stworzonej przez Portable_lab
 - Ćwiczenie umijętności programowania (aplikacji klienckich dla serwera)

Przechwycenie danych z nie zabezpieczonego formularza napisanego w jezyku PHP



Prezentacja jest dostępna na stronie: madwywardugek@leek Pod zakładką Polish Site

Koło naukowe Łączność bez Zakłóceń





Dziękuję za Uwagę

I zapraszam na moją stronę www.ardugeek.pl



jeek