

Hack The Box Netherlands

2nd Wednesday Hack-day meetup

Welcome!

300

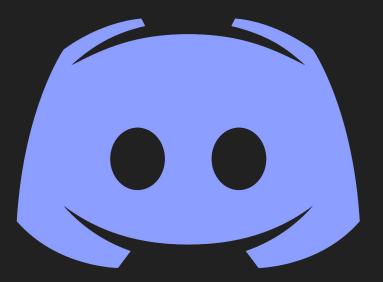
Members in our group

Before we start



- When presentations are going, please mute your microphone (we all love the sound of mechanical keyboards, but it makes it hard to hear people)
- If you have a question, type it in the chat
- Portions of this meetup will be recorded

Discord Community



https://discord.gg/pWh7DSY

Slides and video



You can download these slides from our companion website

https://hackdewereld.nl

What to expect

 Today we have a presentation: Just Enough Reverse Engineering (1 hour)

- After this presentation you will have a choice:
 - Participate in a Reverse Engineering workshop
 - Hack on some machines in our **dedicated lab**



Just Enough Reverse Engineering

to score those Challenge points!

Arjen / credmp



Cyber Security Faculty of NOVI University of Applied Science

KASPERSKY

Certified Malware Reverse Engineering Instructor

Goal for today

- Introduce you to the exciting field of Reverse Engineering
- Show you several tools that form the basis of your RE toolkit
- Teach you a trick or 2 that are used by CTF challenge makers

What is reverse engineering?

the processes of extracting knowledge or design information from anything man-made

- Gives you a great understanding of how computer systems work
- Great fun if you like puzzles
- Allows you to peek "under the hood" of any product

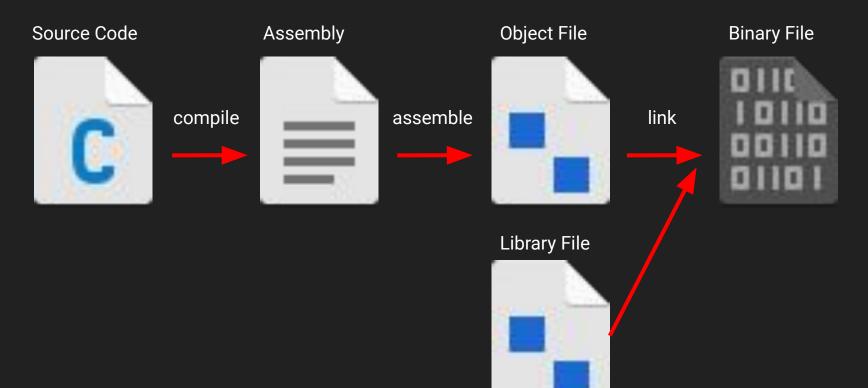
Hack The Box Challenges

- Reversing is one type of challenge
- Today you will learn enough to:
 - Complete the easy challenges
 - Get started on the harder ones

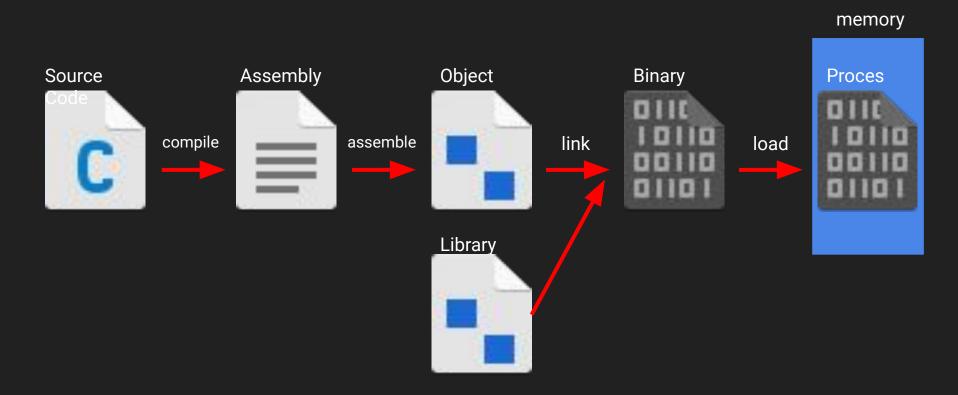
📅 Labs	
Starting Point	
Access	
Machines	
Challenges	
Reversing	10
Crypto	10
Stego	10
Pwn	10
Web	10
Misc	10

What are binaries?

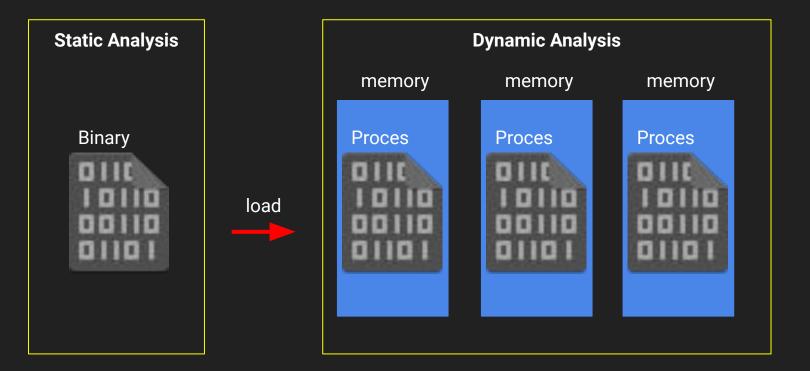
How binaries are created



How binaries are created



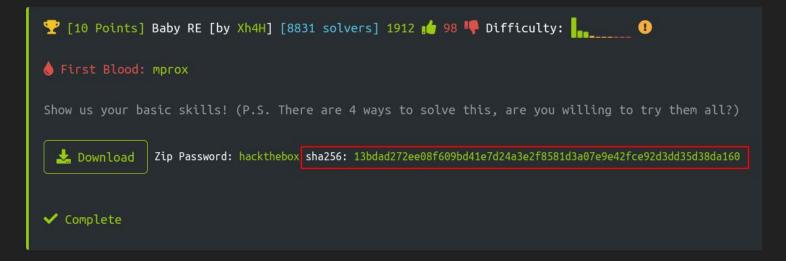
Reverse Engineering Domain



Static Analysis

Do we have the right binary?

- Ensure the zip file is correct by comparing hashes
- Tool: md5sum or sha256sum



Unpacking on Kali

- Use 7z to extract zip archives

```
7z x Baby_RE.zip
                                       7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=en US.utf8,Utf16=on,HugeFiles=on,64 bits,1
Scanning the drive for archives:
1 file, 2885 bytes (3 KiB)
Extracting archive: Baby RE.zip
Path = Baby_RE.zip
Type = zip
Physical Size = 2885
```

Enter password (will not be echoed): Everything is Ok

Tool: file

- used to determine filetype
- with binaries it shows so much more!
 - Executable type (ELF, PE, etc)
 - Architecture
 - Endianess
 - Statically or Dynamically linked

> file baby

baby: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=25adc5 3b89f781335a27bf1b81f5c4cb74581022, for GNU/Linux 3.2.0, not stripped

Tool: strings

- list printable strings in a binary
- allows you to determine program functions
- will show a lot of small strings as well, so need to filter

> strings -n 10 headache /lib64/ld-linux-x86-64.so.2 cxa finalize libc_start_main GLIBC_2.2.5 ITM deregisterTMCloneTable __gmon_start__ _ITM_registerTMCloneTable []A\A]A^A a15abe90c112d09369d9f9da9a8c046e Initialising Enter the key: Login Failed! Login success! ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0: HTB{not so easy lol} GCC: (Debian 8.3.0-19) 8.3.0

Is it packed? (Tool: upx)

- Packing "hides" the actual binary inside the binary
- It is only loaded at runtime and you can not examine it without unpacking
- Most popular packer is **upx**
- upx -I <filename> will show if the binary is packed

Copyrigh	t (C) 1996 - 2	2018	
erhumer,	Laszlo Molnar	& John Reiser	r Aug 26th 2018
Ratio	Format	Name	
d to unpac	k code is by usin	ng memory dun	
32.21%	linux/amd64		
	Copyrigh erhumer, Ratio	Copyright (C) 1996 - 2 erhumer, Laszlo Molnar Ratio Format	t to unpack code is by using memory dur

Disassembler







IDA Pro / Freeware

- Commercial tool from Hex-Rays
- Loved in the industry, but very expensive!
- Comes with a debugger

File Edit Jump Search View Debugger	Options Windows Help
😸 🖬 : er v er v : 🎭 🎭 🗛 🖡	🔊 : 🖾 👄 : 🏦 🏦 🖈 🖈 📽 🗶 : 🕨 💷 💶 Local Linux debugger 🗹 🧌 💽 : 🗊 🊏 🏠
:	
📃 Library function 📃 Regular function 📕 Instru	ction 📃 Data 📕 Unexplored 📒 External symbol
📝 Functions window 🛛 🛛 🖉	🛛 IDA View-A 📮 🗿 Hex View-1 🗉 🖪 Structures 🗉 🗒 Enums 🗉 📆 Imports 🗉 🗺 Exports 🖬
Function name	
_init_proc	: Attributes: bp-based frame
📝 sub_D50	; intcdecl main(int argc, const char **argv, const char **envp)
🗾 _popen	public main main proc near
std::runtime_error::runtime_error(char const	and a state of the
<pre>Image: Image: Imag</pre>	var.40= byte ptr -40h var.18= qword ptr -10h
storcourt.basic_string <char,stor.char_ba< td=""><td></td></char,stor.char_ba<>	
Zcxa_atexit	push rbp mov rbp.rsp
std::operator<< <char,std::char_traits<char>,</char,std::char_traits<char>	push rbx
std::cox11::basic_string <char,std::char_tra< td=""><td>sub rsp, 30h mov rsx, fsi28h</td></char,std::char_tra<>	sub rsp, 30h mov rsx, fsi28h
I std::operator<< <std::char_traits<char>>(std:</std::char_traits<char>	mov [rbp+var_15], rax
🗾 std::ostream::operator<<(std::ostream & (*)(s	xor eax, eax mov edi, 0 ; uid
stack_chk_fail	call _setuid
<pre>std::cox11:basic_string<char,std::char_tra f="" pre="" std::ios_base::init:init(void)<=""></char,std::char_tra></pre>	mov edi, 0 ; gid call setuid
J statios_base:init:init(void)	lea rsi, aHardwareInfo ; "Hardware Info
Zcxa_throw	<pre>lea rdi_Z514coutgeGLBCX_3.4 call Z514s511cher traitsEER511basic ostreamIcT E55 PKc ; std::operator<<<std::char traits<char="">>(std::basic ostream<char,std::char pre="" traits<char<=""></char,std::char></std::char></pre>
Unwind_Resume	nov rdx, rax
📶 _setgid	mov rax, cs: ZSt4endlIcStllchar traitsIcEERStl3basic_ostreanIT_T0_ES6_ptr mov rsi. rax
📝 _setuid	<pre>mov rdi, rdx call ZNSolsEPFRSOS E ; std::ostream::operator<<(std::ostream & (*)(std::ostream &))</pre>
cxa_finalize	lea rax, (rbp+var_40)
✓ _start	lea rsi, alshushort; "lshw -short" mov rdi, rax
deregister_tm_clones	call _Z4exec85cxx11PKc
[Z] register_tm_clones [Z]do_global_dtors_aux	lea rax, [rbp+var_40] mov rsi, rax
frame_dummy	lea rdi, ZSt4coutpeGLIBCXX 3 4
Z4execB5cox11PKc	callZStlsIcStllchar_traitsIcESaIcEERStl3basic_ostreamIT_T0_ES7_RXNESTcxxll12basic_stringIS4_S5_T1_EE ; std::operator<< <char_traits<char_traits<char_traits<char_traits< th=""></char_traits<char_traits<char_traits<char_traits<>
7 main	mov rax, cs: Z5t4endlIcStilchar traitsIcEERSti3basic ostreamIT T0 E56 ptr
static_initialization_and_destruction_0(in	mov rši, raz
_GLOBALsub_IZ4execB5cxx11PKc	call ZNSOLSEPFRSOS E ; std::ostream::operator<<(std::ostream & (*)(std::ostream &))
Std::array <char,128ul>::size(void)</char,128ul>	Laa rax, [rbp+var_40] mov rdi, rax
std::move<_IO_FILE *&>(_IO_FILE *&&&) std::move <int (*&)(_io_file="" *)="">(int (*&)(_IO_FILE *)>(int (*&)(IO_FILE *)>(int (*&)(IO_F</int>	callNSt7_cxx1112basic_stringIcSt11char_traitsIcESaIcEED1Ev ; std::_cxx11::basic_string <char,std::char_traits<char>,std::allocator<char>>::>basi</char></char,std::char_traits<char>
std::move <int (~&)(_io_file="" -)="">(int (~&)(_IO</int>	lea rsi, adiskinfo ; "===================================
	callZStlsIStllchar_traitsIcEERStl3basic_ostreamIcT_ESS_PKc ; std::operator<< <std::char_traits<char>>(std::basic_ostream<char,std::char_traits<char< td=""></char,std::char_traits<char<></std::char_traits<char>
A Graph overview	mov rdx, rax mov rdx, sz. ZsteendLicStlichar_traitsIcEERStl3basic_ostreamIT_T0_ES6_ptr
	mov rsi, rax
	<pre>mov rdi, rdx call _ZNSolsEPFRSos_E ; std::ostream::operator<<(std::ostream & (*)(std::ostream &))</pre>
	lea rax, [rbp+var.40] lea rsi, afdiskt. ; =fdisk -l=
	lea rsi, afdiski, ; "fdisk -l" mov rdi, rax
	08.00% (-29,38) (23,608) 00001192 00000000000192: main+49 (Synchronized with Hex View-1)
Output window	
Propagating type information	
Function argument information has been propag	ated

Ghidra

- Developed by the NSA
- Has been open sourced
- Not the most intuitive, but very powerful
- Comes with a **decompiler**!!!!
- But sadly no debugger yet....

Program Trees	×	Using: MiProgram.exe -	(6 addresses se	riected) X	A Function Graph - B JATC Directorial and Director	×	Py Decompile: @_RTC_OredStadVa_ 😏 🔯 🚱 💌 3
Constant Type Mana	* 3 ×	00411d04 00411d07 00411d08	MOV PUSH PUSH PUSH CALL ADD MOV CMP JZ MOV PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH	17 - 10 - 10 - 10 - 10 - 10 - 10 - 10 -			<pre>do (</pre>

Radare2 / Cutter

- Libre and portable reverse engineering framework
- More than just a disassembler
- Cutter (the GUI) now includes the **Ghidra decompiler**
- Features a **debugger** (Beta)

lag name or address here	View Windows Debug Help	
	▶ ▼ Type flag name or address here	
© © Strings	© © Strings	
Offset Na Address String Type Length Size Section		Size Section
Occording in the second interval interva	tim_clones 4 0	a. North 131 rodstas 131 rodstas 131 rodstas 131 rodstas 131 rodstas 5 eh frame 6 e. 7 strabs 21 strabs 21 strabs 22 strabs 31 strabs 32 strabs 31 strabs 32 strabs 31 strabs 32 strabs 33 strabs 34 strabs 35 strabs 36 strabs 37 strabs 38 strabs 39 strabs 39 strabs 31 strabs 31 strabs 31 strabs 31 strabs 31 strabs 32 strabs 33<

My advice

- When you get started, use **Ghidra**
- Examine the other tools once you finish the challenges (IDA / Cutter)
- Invest some time in learning 1 of these tools very well

Dynamic Analysis

Tracing

- Tool: Itrace
 - Traces library calls performed by the binary
 - C functions such as **puts, fgets, strcmp**

```
arjen@kali:~/Downloads$ ltrace ./secret_
__libc_start_main(0x4008fd, 1, 0x7ffe184
time(0)
srand(0x5ec2f2f0, 0x7ffe1844b7d8, 0x7ffe
rand(0x7fb9795cc740, 0x7ffe1844b6a4, 0x7
fopen("/tmp/secret", "rb")
exit(-2 <no return ...>
+++ exited (status 254) +++
```

- Tool: strace
 - Traces system calls and signals performed by the binary
 - functions related to **memory management, file operations**

Debugger

- Use a debugger to **step through** the program
- You can also manipulate the program by **changing registers**
- Examine / dump memory of the proces
- Use IDA or Cutter or a dedicated debugger

gdb

- The OG debugger under linux, console based
- Extendable, example of this is GEF (<u>https://gef.readthedocs.io/en/master/</u>)
- Extremely powerful
- Hard to learn

<pre>\$f11 : 0x202 \$f12 : 0x000555555555500 → < start+0> xor ebp \$f13 : 0x00057ffffffelB0 → 0x0000000000000 \$f14 : 0x0 \$f14 : 0x0 \$f15 : 0x0003 \$ss: 0x0000 \$es: 0x0000 \$f15: 0x0000 \$g2: 0x0000 \$f2: 0x0000</pre>	
0x00007fffffffe066 +0x00000: 0x00000000000000 - \$rsp 0x00007fffffffe070 +0x0010: 0x0000000000000000 - <_llbc_csu_tht+69> add rbx, 0x1 0x00007fffffffe070 +0x0010: 0x0000000000000000 - <_llbc_csu_tht+69> add rbx, 0x1 0x00007fffffffe070 +0x0010: 0x0000000000000000 - <_llbc_csu_tht+69> add rbx, 0x1 0x00007ffffffe080 +0x0010: 0x0000000000000000 - <_llbc_csu_tht+0> push r15 0x00007fffffffe080 +0x0020: 0x00007555555553070 - <_start+0> xor ebp, ebp 0x00007fffffffe080 +0x0020: 0x000075555555070 - <_start+0> xor ebp, ebp 0x00007fffffffffffe080 +0x0030: 0x000075555555070 - <_start+0> xor ebp, ebp 0x00007fffffffffffe080 +0x0030: 0x000075555555070 - <_start+0> xor ebp, ebp 0x000007ffffffffe080 +0x0030: 0x000075555555000 - <_start+0> xor ebp, ebp	
0x5555555515d cmain+8> lea rax, [rtp+0xen4] # 0x55555556008 0x55555555516d mov 0WORD PTR [rbp-0x8], rax 0x55555555608 0x5555555516d mov 0WORD PTR [rbp-0x8], rax 0x55555555608 0x55555555516d mov Call 0x55555555608 0x555555555508 cputs@plt+6> jmp 0WORD PTR [rbp+0x2fc2] 0x55555558018 0x5555555555080 oputs@plt+6> jmp 0x555555555020 oputs@plt+6> 0x55555555804 oputs@plt+6> jmp 0WORD PTR [rbp+0x2fda] # 0x55555558020 0x55555555804 ofgets@plt+6> jmp 0WORD PTR [rbp+0x2fda] # 0x55555558020 0x55555558046 ofgets@plt+6> putsh 0x1 0x1 0x2	
puts@plt (\$rdl = 0x0000555555556046 → "Insert key:", \$rsl = 0x00007fffffffe188 → 0x00007ffffffe48a → "/home/arjen/Downloads/baby", \$rdx = 0x00007fffffffe198 → 0x00007fffffffe4a5 → "SHELL=/bln/zsh")	
[#0] Id 1, Name: "baby", stopped, reason: SINGLE STEP [#0] 0x5555555516f → main()	

edb - Evan's debugger

- GUI based debugger
- Great alternative to GDB for most cases

File View	Dahua Dh	Ontings 11.1			
	Debug Plugins	Options Help	2		
0 J	⊡_Ľ D 📘				
	:e546 83 ec 0c	sul	esp, 0xc	Registers	\$
	:e549 8d 83 28 e6		eax, [ebx-0x19d8]		
	:e54f 50		sh eax		ASCII "Hello, world!"
	:e550 e8 5b fe ff		ll hello!puts@plt	ECX ff9daa10	
	:e555 83 c4 10		d esp, 0x10	EDX ff9daa34	
	:e558 83 45 e4 01		d dword [ebp-0x1c], 1	EBX 5662ffd8 ESP ff9da9c0	
	:e55c 8b 45 e4		eax, [ebp-0x1c]	EBP ff9da9f8	
	:e55f 3b 06 :e561 7c e3		eax, [esi] 0x5662e546	ESI ff9daa10	
	:e563 b8 00 00 00		eax, 0	EDI 0000000	
	:e568 8d 65 f4		a esp. [ebp-0xc]		
	:e56b 59		ecx	EIP 5662e550	<hello!main+51></hello!main+51>
	:e56c 5b		ebx	C 0 ES 002t	(000000000000000)
	:e56d 5e		esi	P 0 CS 0023	
	:e56e 5d		ebp	A 1 55 002t	(000000000000000)
	:e56f 8d 61 fc		esp. [ecx-4]	Z 0 DS 002t	(000000000000000)
5662	:e572 c3	L ret		5 1 FS 0000	
	:e573 66 90	nor			(0000000f7f500c0)
5662	:e575 66 90	nor	0	DO	
	:e577 66 90	nor		00	
	:e579 66 90	noj		FEL 00000292	(NO,AE,NE,A,S,NP,L,LE)
	:e57b 66 90	noj	0		
				J ST0 empty	0.0
5000					
) = 0x0000000056	62e3b0 <hello!p< th=""><th>outs@plt+0></th><th>ST1 empty</th><th>0.0</th></hello!p<>	outs@plt+0>	ST1 empty	0.0
0x5662e3b0) = 0x000000056				
0x5662e3b0				ST1 empty	0.0
0x5662e3b0 puts(<0x00) = 0x000000056			Bookmarks	0.0 0.0 Registers
0x5662e3b0) = 0x000000056			ST1 empty ST2 empty Bookmarks Stack	0.0 0.0 Registers
0x5662e3b6 puts(<0x00 Pata Dump) = 0x000000056	"Hello, world!		ST1 empty ST2 empty Bookmarks Stack ff9d:a9c0 566	0.0 0.0 Registers 2e6000 . wbV ASCII *Hello,
0x5662e3b6 puts(<0x00 Pata Dump) = 0x0000000056 00000005662e600> 000005662e000-0x00	"Hello, world!	·*) **	ST1 empty ST2 empty Bookmarks Stack	0.0 0.0 Registers 226600
0x5662e3b6 outs (<0x06 ata Dump 0x0000 5662 : e000	0 = 0x000000056 0000005662e600> 0000005662e000-0x00 7f 45 4c 46 01 01	"Hello, world!	• 00 00 00 00 .ELF	ST1 empty ST2 emntv Bookmarks Stack ff9d:a9c0 566 ff9d:a9c4 f7f ff9d:a9c4 000	0.0 0.0 Registers 226600
0x5662e3b0 puts(<0x00 hata Dump € 0x0000 5662:e000 5662:e000) = 0x000000056 0000005662e600> 0000005662e000-0x00 7f 45 4c 46 01 01 03 00 03 00 01 00	"Hello, world!	(*) 0 00 00 00 00	ST1 empty ST2 emntv Bookmarks Stack ff9d:a9c0 566 ff9d:a9c4 f7f ff9d:a9c4 000	0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
ata Dump 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x000 0x00 0x000 0x000 0x000 0x000 0x000) = 0x000000056 00000005662e600> 0000005662e000-0x00 7f 45 4c 46 01 01 03 00 03 00 01 00 14 17 00 06 00 00	"Hello, world!	• 00 00 00 00 ELF	ST1 empty ST2 empty Bookmarks Stack ff9d:a9c0 566 ff9d:a9c0 566 ff9d:a9c6 566 ff9d:a9c6 566 ff9d:a9c6 566 ff9d:a9c6 566 ff9d:a9d6 566	Registers
ata Dump 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x00 0x000 0x000 0x000 0x000 0x000 0x000 0x000 0x00 0x000 0x000 0x000 0x000 0x00 0x00 0x000 0x00 0) = 0x000000056 0000005662e600> 0000005662e000-0x0(7f 45 4c 46 01 01 03 00 03 00 01 00 41 17 00 00 00 00 40 00 1c 00 65 00	"Hello, world!	• • • • • • • • • • • • • • • • • • •	Stack Stack 1993:39:49:566 1993:39:49:49:566 1993:39:49:49:566 1993:39:49:49:49:566 1993:39:49:49:49:49:49:566 1993:39:49:49:49:49:49:49:49:49:49:49:49:49:49	Registers 2e680 wbV ASCII "Hello, 2e680 x 2e680 x 2e535 5WV return to 0x00 2e37c 0 2frd8 (DDV) daac ar.
0x5662e3b0 puts (<0x00 tata Dump) = 0x000000056 0000005662e600> 000005662e000-0x00 7f 45 4c 46 01 01 30 00 03 00 01 00 41 17 00 60 00 00 41 00 00 00 62 0 01	"Hello, world! 00000005662f000 01 00 00 00 00 00 00 00 24 00 20 00 00 34 00 20 00 00 34 00 00 00 00 34 00 00 00		ST1 empty ST2 empty Bookmarks Stack ff9d:a9c0 566 ff9d:a9c4 566 ff9d:a9c4 566 ff9d:a9c4 566 ff9d:a9d4 566 ff9d:a9d6 ff9d:a9d6 ff9d:a9d6 ff9	Registers
0x5662e3b0 puts (<0x000 0x0000 5662:e010 5662:e010 5662:e010 5662:e020 5662:e030 5662:e030 5662:e030) = 0x000000056 0000005662e000- 0000005662e000-0x0(7f 45 4c 46 01 01 03 00 03 00 01 00 41 17 00 00 00 00 40 00 1c 00 65 00 41 00 00 00 60 30 00 00 00 03 00	"Hello, world!	••••••••••••••••••••••••••••••••••••	511 enpty 512 ennty Bookmarks Stack 1194:a9c8 566 1194:a9c8 866 1194:a9c8 800 1194:a9c8 800 1194:a9c8 800 1194:a9c8 800 1194:a9c8 800	Registers 2e660bV ASCII "Hello, 00000 2e37c 2e37c 2e37c 2e37c 2e37c 00000 00001
0x5662e3b0 puts (<0x000 tata Dump 0x0000 5662:e010 5662:e010 5662:e030 5662:e030 5662:e030 5662:e030	0 = 0x000000056 0000005662e000-0x00 7f 45 4c 46 01 01 03 00 3 00 61 00 1d 10 1c 00 00 20 01 1d 00 1c 00 06 20 01 94 00 00 06 03 00 40 10 00 06 13 00	"Hello, world!	••••••••••••••••••••••••••••••••••••	ST1 empty SD0kmarks Bookmarks Stack ff9d:a9c0 566 ff9d:a9c4 800	Registers 2c600 .wbV ASCII "Hello, 2c600 .wbV ASCII "Hello, 2c600 2c355 5wV return to 0x0 2c37c 100V 2c37c 100V 00000 00000 00000
0x5662e3b0 puts (<0x00 bata Dump	0 = 0x000000056 0000005662e000-0x00 77 45 4c 46 01 01 30 00 30 06 10 34 17 00 00 00 10 40 01 c 00 65 00 34 00 00 00 20 01 40 00 00 00 30 40 00 00 03 00 54 01 00 00 13 00 10 00 00 06 10	"Hello, world 00000005662f000 01 60 00 00 00 00 60 34 00 20 00 60 34 00 00 00 60 34 00 00 00 60 34 00 00 00 60 34 00 00 00 60 54 00 00 00 60 54 00 00 00 60 50 00 00	••••••••••••••••••••••••••••••••••••	ST1 empty SD2 ennty Bookmarks Stack Stack Stack If94:s9C4 Stack If94:s9C6 Stack If94:s9C6 Stack	Registers 2e660 - ₩bV ASCII "Hello, 2e600 - 4 2e535 - 5bV return to 0x00 2e37C 2frds [DbV daac ar.] 000001 daac ar.]
ata Dump ata Dump 0x0000 5662:e000 5662:e000 5662:e000 5662:e000 5662:e000 5662:e000 5662:e000 5662:e000 5662:e000 5662:e000	• 0x90000005662e600-> 0000005662e0000x00 000005662e0000x00 7f 45 4c 46 01 01 33 00 03 00 01 00 1d 00 1c 00 06 00 1d 00 1c 00 06 03 1d 00 0 0 00 03 10 00 00 01 30 11 00 00 00 03 40	"Hello, world 00000005662f000 81 88 98 98 98 88 89 88 98 98 98 98 89 88 98 98 98 80 88 24 97 88 80 88 24 97 88 80 88 24 97 88	••••••••••••••••••••••••••••••••••••	511 enpty 512 enoty Bookmarks Stack 1190:a920 566 1190:a926 566 1190:a926 966 1190:a926 966 1190:a926 960 1190:a926 900 1190:a926 910 1190:a926 910 1190:a926 119	Registers 2c660 .wbV ASCII "Hello, 2c660 .wbV ASCII "Hello, 2c600 2c535 .bWV return to 0x00 2c17c
ata Dump (<) 0x0000		"Hello, world 000000056625000 61 00 00 00 00 00 00 20 00 00 00 31 00 20 00 00 20 10 00 00 00 20 10 00 00 00 20 10 00 00 00 21 00 00 00 20 10 00 00 00 21 00 00 00 00 00 00	••••••••••••••••••••••••••••••••••••	ST1 enty SD2 enty Bookmarks Stack Stack Stack ST94::s9C8 S66 ST94::s9C8 S79 ST94::s9C8 S79 ST94::s9C8 S66 ST94::s9C8 S79 ST94::s9C8 S79	Registers 226600, WV ASCII "Hello, 22600, VV ASCII "Hello, 22600, VV return to 0x00 27100 [DVV 27105 [DVV 60001 00000 60000
ax5662e3b@		"Hello, world 000000056627000 01 00 00 00 00 00 00 00 00 00 00 01 00 00 01 00 00 00 00 00 00 00 00 00 00	••••••••••••••••••••••••••••••••••••	511 empty 512 emoty Bookmarks Stack 1190:a926 566 1190:a926 956 1190:a926 956 1190:a95	Registers 2e660
ax5662e3b@		"Hello, world 000000056627000 01 00 00 00 00 00 00 00 00 00 00 01 00 00 01 00 00 00 00 00 00 00 00 00 00	••••••••••••••••••••••••••••••••••••	ST1 enty SD2 enty Bookmarks Stack Stack Stack ST94::s9C8 S66 ST94::s9C8 S79 ST94::s9C8 S79 ST94::s9C8 S66 ST94::s9C8 S79 ST94::s9C8 S79	Registers 2e660

CTF Thing to know

XOR

- CTF developers love to XOR flags
- Operation on a binary level
- Output true (1) only when the values differ
- Easily implemented in any programming language (^)

Wrapping up

Where to go from here?

- Reversing challenges on Hack The Box
- <u>https://beginners.re/</u> a guide to reverse engineering examining code and assembly
- <u>https://malwareunicorn.org/#/workshops</u> great workshops you can follow at home

Additional challenges

- Advanced: <u>http://www.flare-on.com/</u> a yearly competition for RE
- <u>http://crackmes.cf/</u> archive of 1000s of crackme challenges

You now know...

- how to perform basic static analysis
 - sha256sum, file, strings, upx, Ghidra

- how to get started with dynamic analysis
 - Itrace, strace,, edb

- how a **XOR** operation works
- you can grab some challenge points!

Thanks! Questions?

Stick around

- There will be 2 breakout rooms
 - 1 to get started reversing
 - 2 to do some hacking on HTB
- **credmp** will host the reversing breakout
- **DutchPyro** will host the HTB hacking breakout

Let us know in the chat where you want to go and what your HTB
 Username is for the dedicated server!

Next Meetup

June 17th (6pm to 9pm)

Workshop

Challenges to try

- BabyRE (easy)
- Exatlon (easy)
- Impossible Password (easy)
- Find the Secret Flag (medium)

Ghidra

- Download from <u>https://ghidra-sre.org/</u>
- Extract in your Downloads directory
- Run ghidraRun to start it

Ghidra first step

- File > New Project
- Give it a name (challenges)
- Click on Finish

	Ghidra: challenges	_ = ×
<u> E</u> ile <u>E</u> dit <u>P</u> roject <u>T</u> ools <u>H</u> elp		
い も男男男男		
Tool Chest		
A 4		
Active Project: challenges		
- 🕝 challenges		
Filter:		2
Tree View Table View		
Running Tools		
		Workspace 🗸
Creating project: /home/arjen/tmp/challeng	es	

Add a binary

- File > Import File
- Select the unpacked challenge file
- Hit Ok in the popup dialog

Format:	Executable and Linking Format (ELF)	- 0
Language:	x86:LE:64:default:gcc	
Destination Folder:	challenges:/	
Program Name:	headache	
	Opti	ons

Details

- Scan it, hit **OK**

	Import Results Summary X			
	Project File Name:	headache		
	Last Modified:	Wed May 20 10:50:33 CEST 2020		
	Readonly:	false		
	Program Name:	headache		
	Language ID:	x86:LE:64:default (2.9)		
	Compiler ID:	gcc		
	Processor:	x86		
	Endian:	Little		
	Address Size:	64		
	Minimum Address:	00100000		
		_elfSectionHeaders::000000ff		
	# of Bytes:	11073		
	# of Memory Blocks:	7		
	# of Instructions:	0		
		78		
		46		
	# of Symbols:	66		
	# of Data Types:	21		
# of Data Type Categories: 2 Created With Ghidra Version:9.1.2				
	Date Created:	Wed May 20 10:50:32 CEST 2020		
	ELF File Type:	shared object		
	ELF Original Image Base:	0x0		
	ELF Prelinked:	false		
	ELF Required Library [0]			
	Executable Format:			
	Executable Location:	/home/arjen/Challenges/reversing/headache/headache		
	Executable MD5:	3f33f0eaf07d1f8fcb8b007ce9ffcfc1		
	Executable SHA256:	2d4b1425ad9c8dc78eb37a3647d062b356829a876a770ecd36154f807b7d4a86		
	FSRL:	file:///home/arjen/Challenges/reversing/headache/headache?MD5=3f33f0eaf07d1f8fc		
	Relocatable:	true		

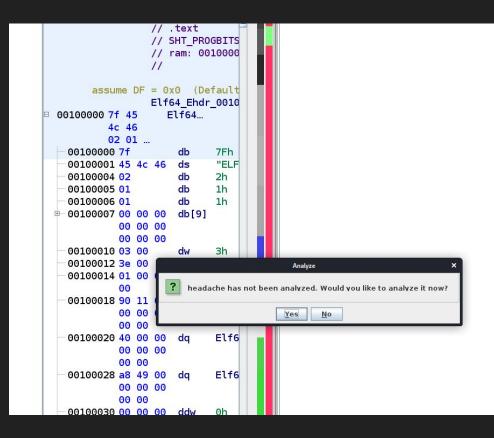
Additional Information-

4

----- Loading /home/arjen/Challenges/reversing/headache/headache -----[libc.so.6] -> not found ----- [headache] Resolve 23 external symbols -----Unresolved external symbols which remain: 23 •

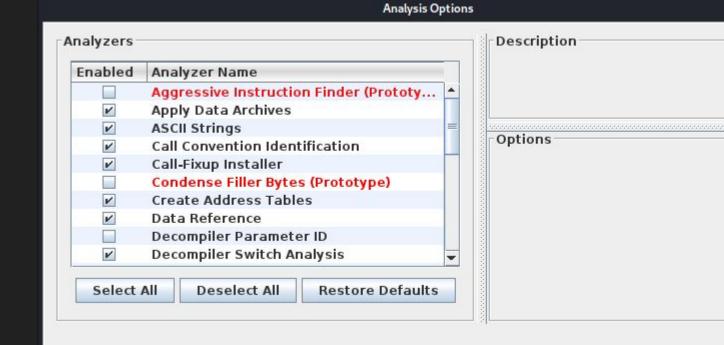
Start analyzing

- Double click on the binary
- Select Yes to analyse your binary



Analyze!

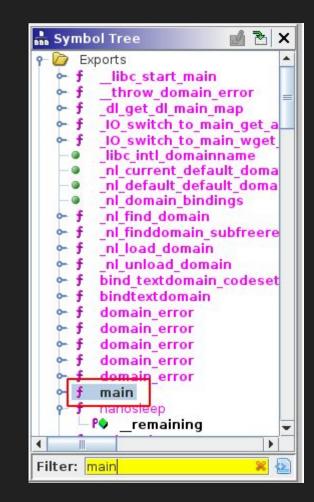
- Hit Analyze





Find the entrypoint

- In the Symbol Tree expand the Exports
- Search for something like entry or main
- Start your quest here.



Start your journey...