

Capture The Flag

An Introduction



Me

- ◉ Jordan Wiens

✉ jordan@psifertex.com

🐦 @psifertex

- ◉ Day: Work for The Man

- ◉ Night: Hack for The Fun



Outline

- What's CTF?
- Why CTF?
- Playing CTF?
- Running CTF?

What's CTF?

Capture the _ _ _ _



Jargon File

flag

key

SLA

service

binary

quals

prequals

Hacker Games
For 2-4 Players

HACKER

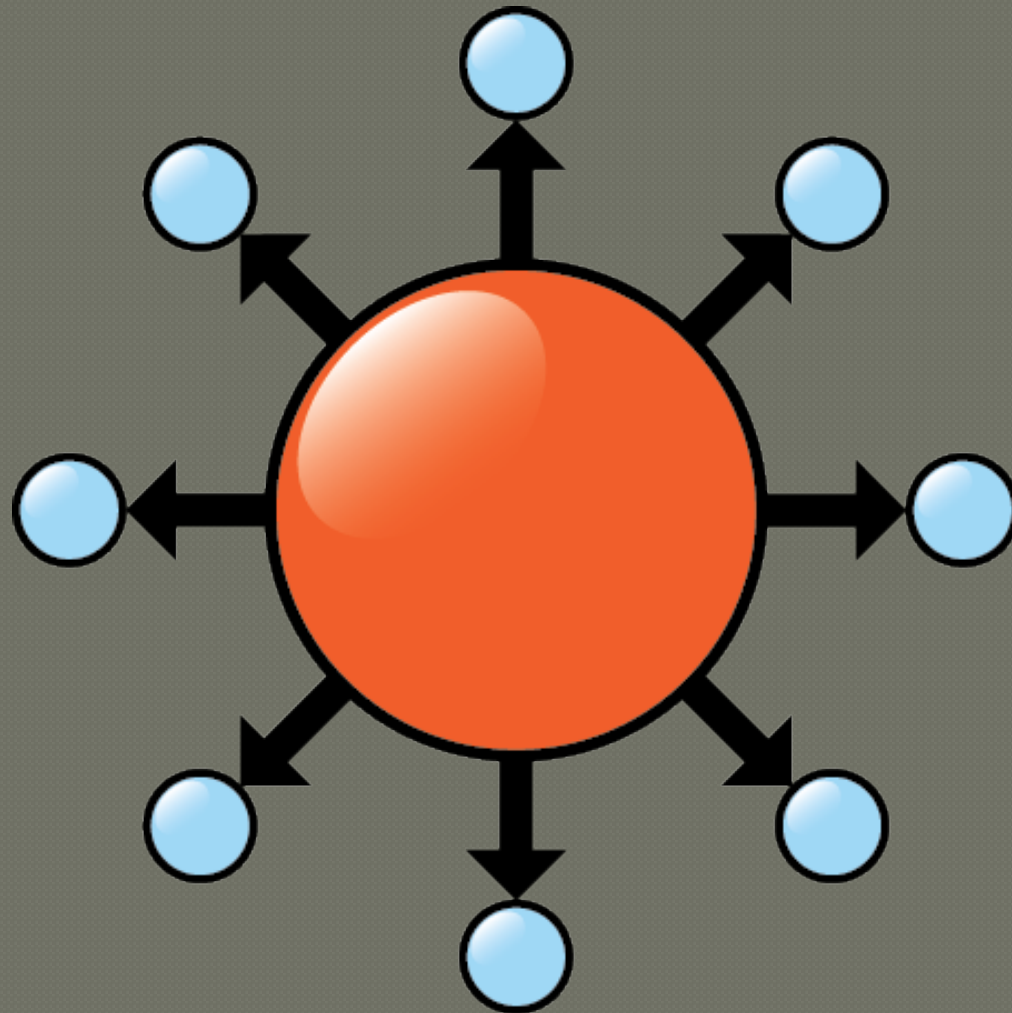
Tracker



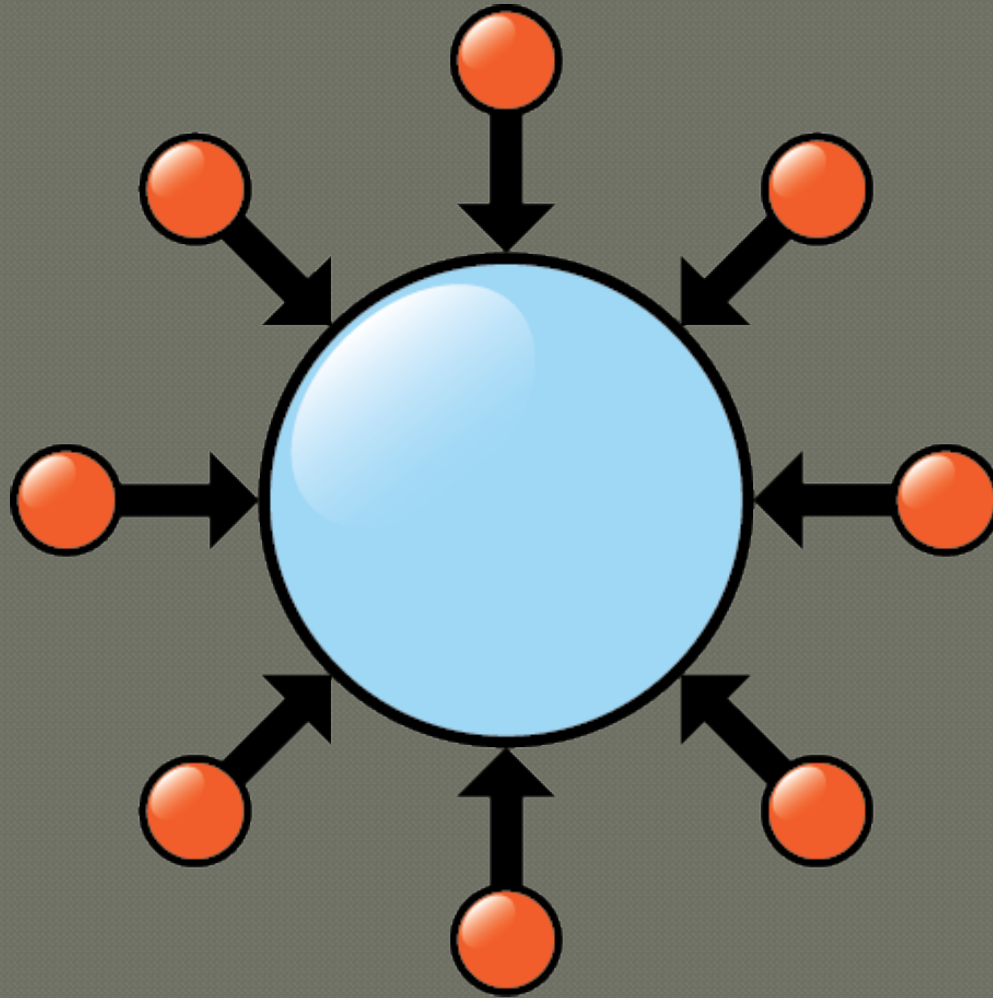
A Strategic Manhunt
where Hot Clues and
Hacker Profiles Help You
Capture a Cyber Criminal



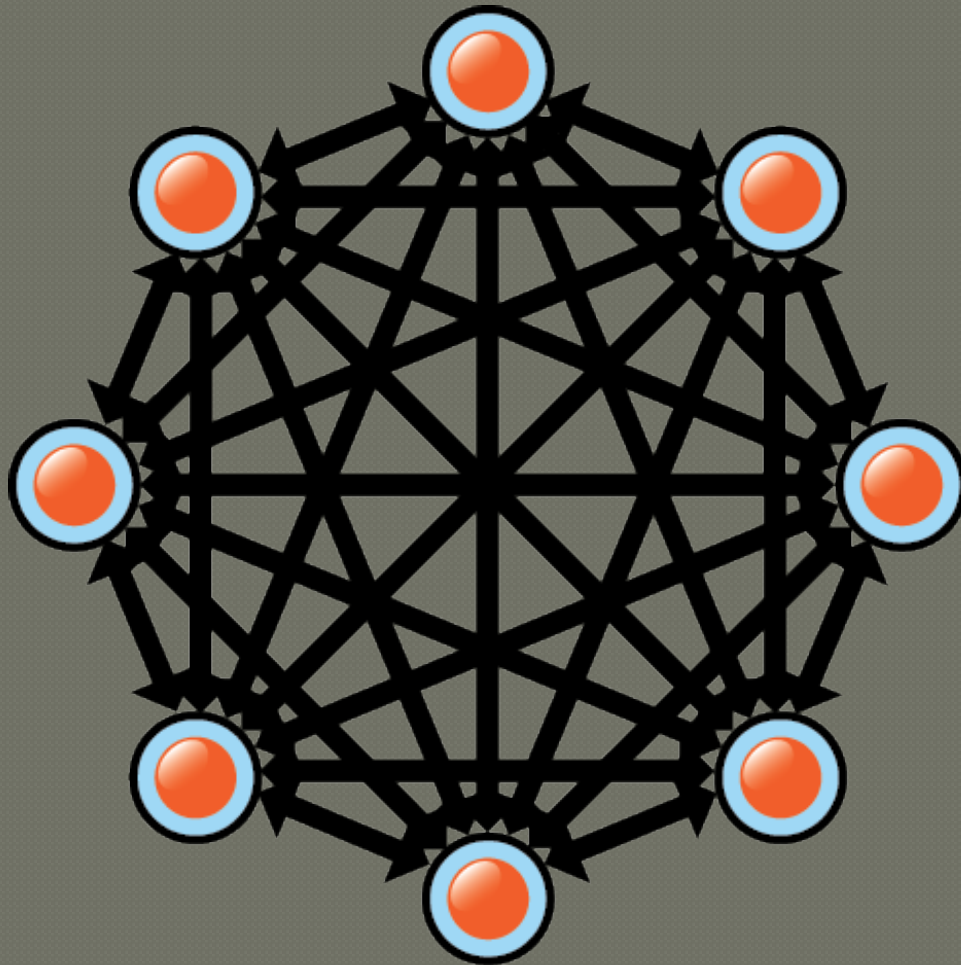
BLUE TEAM



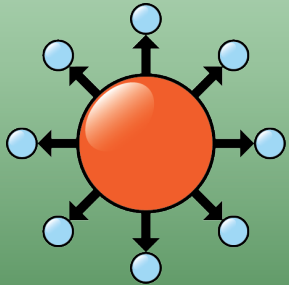
RED TEAM



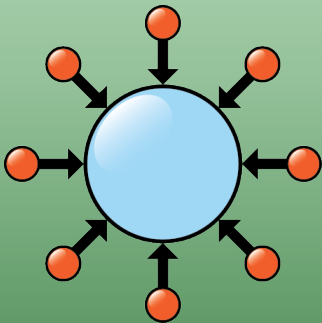
FULL SPECTRUM



CTFs

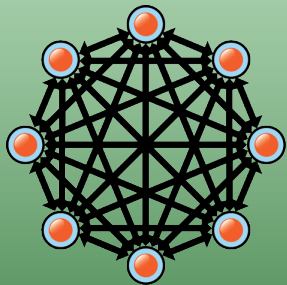


Collegiate Cyber Defense Competition (CCDC)
Cyber Defense Exercise (CDX)



DEF CON Quals
Ghost in the Shellcode
PlaidCTF
CSAW

Kommand & Kontroll
RuCTF Quals
Nuit du Hack Quals
Hack.lu

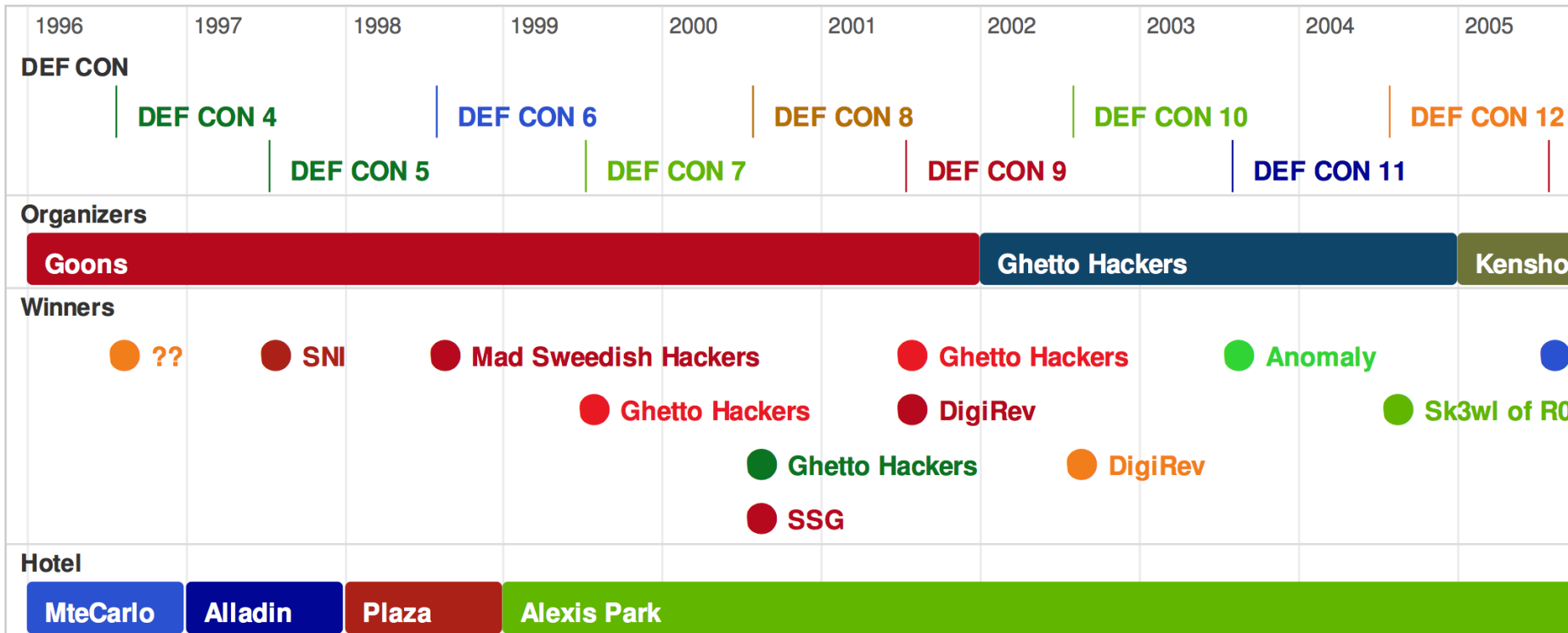


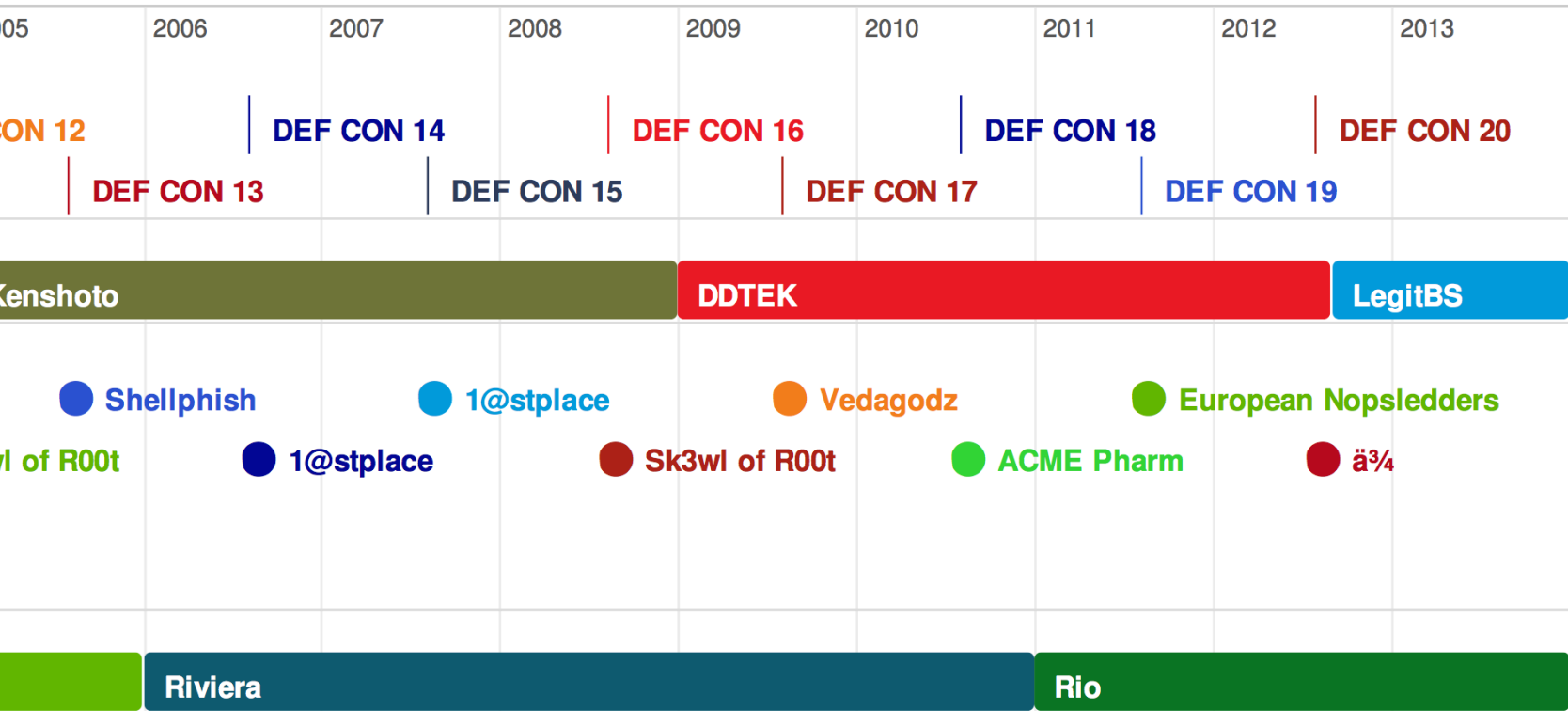
DEF CON Finals
CODEGATE YUT
RuCTF Finals

Nuit du Hack Finals
rwthCTF
iCTF

DEF CON CTF History

- ◉ <http://preceden.com/timelines/62839>





“Jeopardy” (Defcon Quals)

Score: 0000

Logout

Pursuits Trivial	Crypto Badness	Packet Madness	Binary L33tness	Pwntent Pwnables	Forensics
100	100	100	100	100	100
200	200	200	200	200	200
300	300	300	300	300	300
400	400	400	400	400	400
500	500	500	500	500	500

Howie doit?

• [file](#)

I pwn3d U

Leaders

1. sk3wlm4st3r (5400)
2. VedaGodz (5000)
3. Sexy Pwndas (5000)
4. PLUS (4500)
5. Shellphish (4500)
6. Song of Freedom (4400)
7. lollerskaterz dropping from roflcopters (4300)
8. Underminers (4300)
9. Routards (4200)
10. WOWHACKER (3800)
11. Sapheads_ (3700)
12. sutegoma (3400)
13. CLiP (3300)
14. pebkac (3300)
15. ACMEPharm (3300)

Why CTF?

Jobs

Skillz

\$\$

Cred



Sold for
\$3550.01
at auction

Fun!

2006 Quals – Reversing 400

The screenshot shows the IDA Pro interface. On the left, the 'Functions window' lists various functions, with 'sub_804845C' selected. The main window displays the assembly code for 'sub_804845C' in the 'Pseudocode-A' view. The code includes a comment about attributes, a procedure declaration, and several instructions: pushing the frame pointer, moving it to ESP, adjusting the stack pointer, pushing an offset, calling _puts, adjusting ESP, pushing a status value, and calling _exit.

```
Functions window
Function name
_ _init_proc
_ _puts
_ __libc_start_main
_ _exit
_ _calloc
_ _fwrite
_ __gmon_start__
_ start
_ sub_80483BC
_ sub_80483E0
_ sub_8048414
_ sub_804843C
_ sub_804845C
_ sub_80484BC
_ sub_8048518
_ sub_8048520
_ _term_proc
_ puts
_ __libc_start_main
_ exit
_ calloc
_ fwrite

IDA View-A
Pseudocode-A
Hex View-A
Structures
Endu

; Attributes: noreturn bp-based frame

sub_804843C proc near
push    ebp
mov     ebp, esp
sub     esp, 8
sub     esp, 0Ch
push    offset s          ; "Usage: encoder <string>"
call    _puts
add     esp, 10h
sub     esp, 0Ch
push    0FFFFFFFh        ; status
call    _exit
sub_804843C endp
```

2006 Quals – Reversing 400

```
1 void __cdecl sub_8049688(int a1)
2 {
3     size_t v1; // kr04_4@1
4     int v2; // [sp+20h] [bp-18h]@1
5     char v3; // [sp+24h] [bp-14h]@1
6     signed int i; // [sp+28h] [bp-10h]@1
7     void *ptr; // [sp+2Ch] [bp-Ch]@1
8
9     v2 = *(_DWORD *)(a1 + 4);
10    v3 = 0;
11    v1 = strlen(*(const char **)(a1 + 4)) + 1;
12    ptr = calloc(1u, v1 - 1 + 2);
13    for ( i = 0; i < (signed int)(v1 - 1); ++i )
14    {
15        *((_BYTE *)ptr + i) = 8 * *(_BYTE *)(v2 + i);
16        *((_BYTE *)ptr + i) |= v3;
17        if ( (unsigned int)(i % 8) <= 7 )
18            JUMPOUT(__CS__, dword_8048584[i % 8]);
19        v3 = *(_BYTE *)(v2 + i) >> 5;
20    }
21    *((_BYTE *)ptr + i) = v3;
22    fwrite(ptr, v1, 1u, stdout);
23    exit(0);
24 }
```


2006 Quals – Reversing 400

```
$ hexdump key.enc
```

```
0000000 f9ef 0942 1aa3 f743 8b8c 22bb c22a 14a3
```

```
0000010 0003
```

2006 Quals – Reversing 400

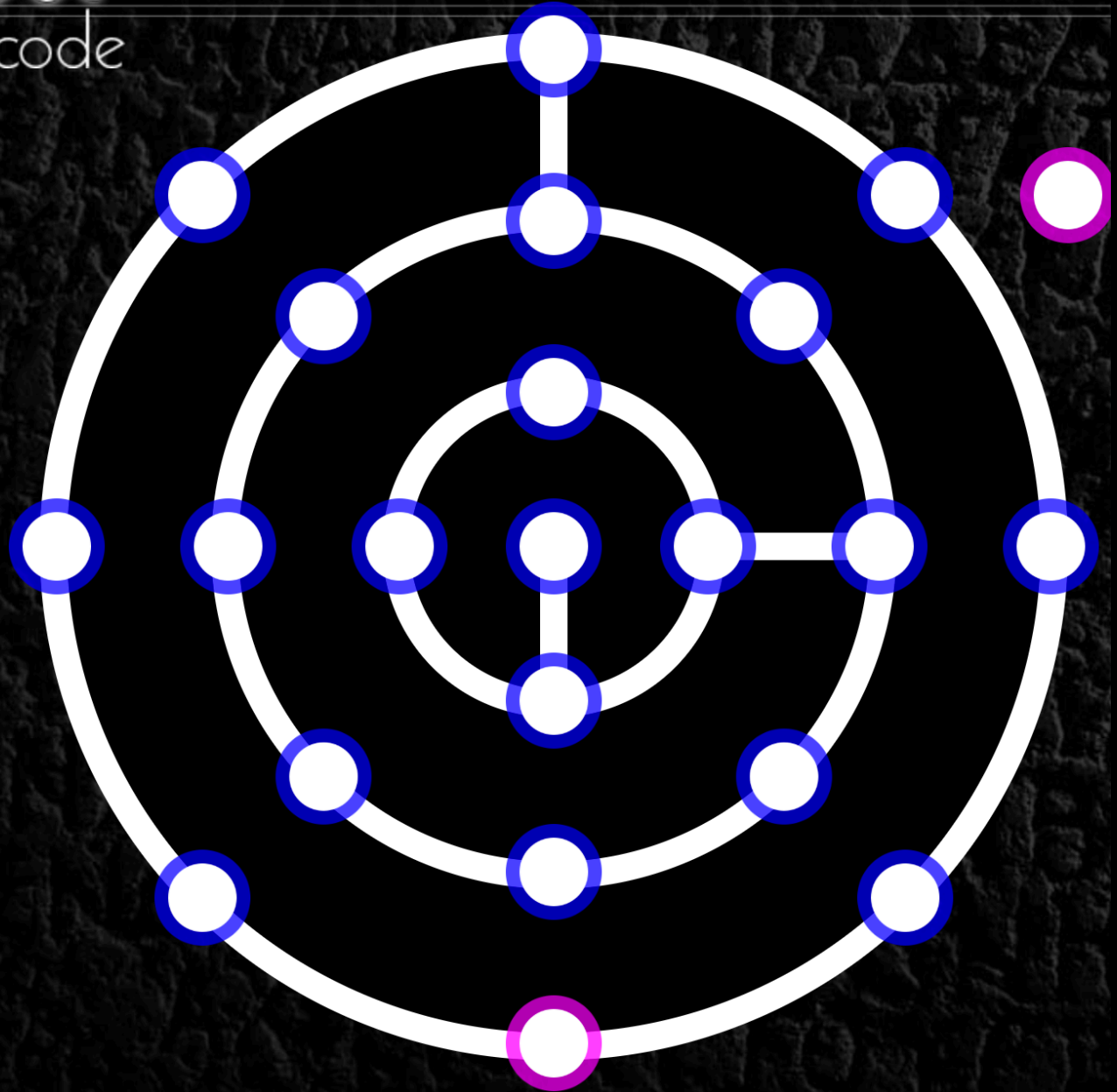
```
#!/bin/bash
for c1 in {a..z}; do
    for c2 in {a..z}; do
        echo -e "$c1$c2\t" `./encoder \
"$c1$c2"|hexdump|head -1`
    done
done
```

ghost

in the shellcode

Top 10		
#	Team Name	Score
1	PPP ‡	4550
2	More Smoked Leet Chicken	2750
3	Dragon Sector	2550
4	TheSamurai ‡	2500
5	pwnies	2450
6	TracerTea	2400
7	0x8f	2051
8	European Nopsled Team	2000
9	GoN	1950
10	Eindbazen	1950

‡ is at ShmooCon.



What the hell is keming?

1. Run “file”

```
$ file 35e25782a7b3b88409e58756e63c40c2.bin
35e25782a7b3b88409e58756e63c40c2.bin: XZ compressed data
```

What the hell is keming?

1. Run “file”
2. **Read spec (RFC-1952)**

“A gzip file consists of a series of ‘members’ (compressed data sets). The format of each member is specified in the following section. The members simply appear one after another in the file, with no additional information before, between, or after them.”

What the hell is keming?

1. Run “file”
2. Read spec (RFC-1952)
3. **Extract / re-arrange**

Exercise for the reader. (python, binwalk, shell script)

What the hell is keming?

1. **Run “file”**
2. Read spec (RFC-1952)
3. **Extract / re-arrange**
4. **GOTO 1**

```
$ file output
output: POSIX tar archive (GNU)
$ tar -xvf output
keming/
keming/index.html
keming/pronoun.woff
keming/preposition.woff
keming/adjective.woff
keming/interjection.woff
```

What the hell is keming?

1. Run “file”
2. **Read spec (<http://w3.org/TR/WOFF>)**
3. Extract / re-arrange
4. GOTO 1

Exercise for the reader.

Playing

Skills...

- x86/MIPS/ARM/PPC/Atmel
- Reverse Engineering
- Binary Exploitation
- File System Forensics
- File Format Forensics
- Cryptography
- Web App Sec
- Hacker Trivia
- Emulation/Virtualization
- Custom compression
- Programming (scripting and the real deal)
- PHP “Phun”
- Binary Protection Mechanisms
- Formal Methods
- Network Protocol Analysis
- Shellcode Tricks
- Number Systems
- Bizarre Encodings

The one true secret
to success:

TRYING

Resources

● Calendars

- <http://captf.com/calendar>
- <http://ctf.forgottensec.com/wiki>
- <http://ctftime.org/ctfs/>

Resources

● Archives

- <http://captf.com/>
- <http://shell-storm.org/repo/CTF/>
- <http://ctftime.org/event/list/past/>

Resources

● Practices

- <http://captf.com/practice-ctf/>
- <http://www.wechall.net/>
- <http://ctf.forgottensec.com/wiki>

Resources

● Videos

- Hacker Joe (<http://youtu.be/6e4kJB4cthA>)
- Psifertex(<http://youtu.be/okPWY0FeUoU>)
- Chris Eagle (<http://vimeo.com/29689138>)
- Arpaia (<http://vimeo.com/30141771>)
- ShmooCon (<http://youtu.be/c9Rc6DjYJr8>)

Running

Schlock Mercenary



Schlock Mercenary Wiki

The Seventy Maxims of Maximally Effective Mercenaries

On the Wiki
Wiki Activity
Popular pages
Random page

[Edit](#) [Comments](#) 2

The *Seventy Maxims of Maximally Effective Mercenaries* is a popular handbook in the universe. The book's maxims are often quoted by [Tagon](#), as well as other characters. maxims found in *Schlock Mercenary*, ordered by maxim number.

1. Pillage, then burn.^[1]
2. A Sergeant in motion outranks a Lieutenant who doesn't know what's going on.^[2]
3. An ordnance technician at a dead run outranks everybody.^[3]
4. Close air support covereth a multitude of sins.^[4]
5. Close air support and friendly fire should be easier to tell apart.^[5]



<http://schlockmercenary.com/>

The Many Maxims of a Maximally Effective CTF

<http://captf.com/maxims.html>

Maxims

1. We hack for fun, not for frustration.
2. The scoring mechanism should always be the easiest challenge.
3. Solutions might be a surprise, but recognizing when you have one shouldn't be.
4. When the next step requires a leap of faith, be sure to include a bridge.
5. An homage honors, but duplication doesn't.

Maxims

6. Learners always win even when winners don't learn.
7. Your point estimates are exactly that until calibrated.
8. Never rely on the survival of a vulnerable server.
9. Competitors are more clever than you, they also have more time.
10. Learning starts where prior knowledge ends.

Questions?

THANKS:

family, friends, CTF mates past and present, HackUCF, b-sides organizers!

SLIDES:

<http://captf.com/intro>

Bonus content!

Other things

- Team Organization
- Culture of the game
 - Good / bad
- Strategy
 - Sleep
 - Play the organizers
 - “scrum”