Hacking like in the movies

Insomni'hack 2015 teaser

writeups

INSOMNI'HACK

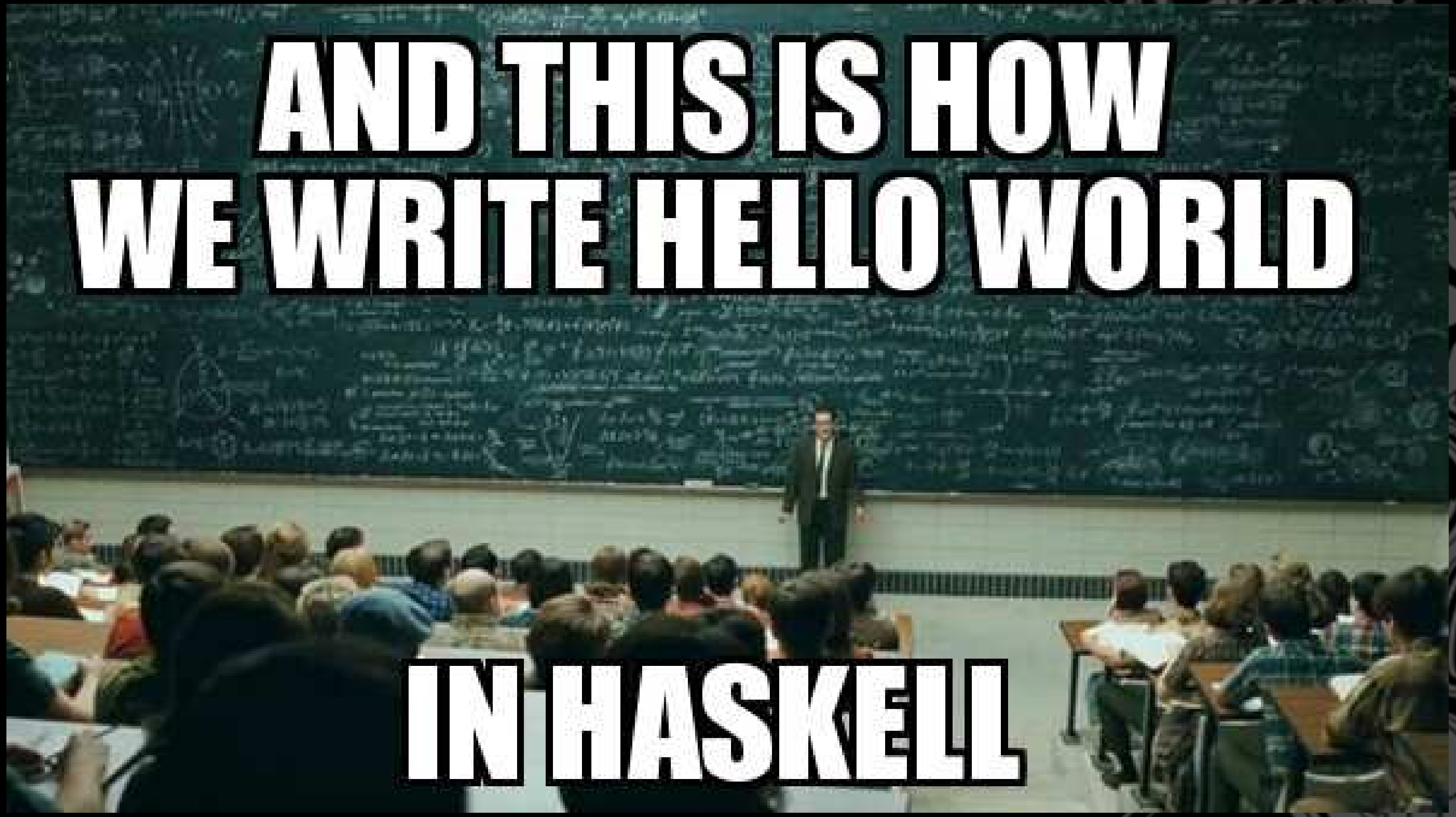
Intro | Teaser

- Main event CTF is open to all – no quals
- Invite first few teams to conference
- Longer than the main CTF (36h)
  ⇨ Fewer but harder tasks
- 3 pwnables, 1 reverse, 1 web

Intro | Scoreboard

- Scoreboard running on Haskell
- NodeJS at the finals
- Very optimized
- Unreadable ;)
- Infra on AWS

- The Hipster's choice!



Pwning|SH1TTY (finals)

- Intended for the Teaser
- Timing FAIL
- Moved to the finals, made easier
- Solved a few minutes after the CTF end ☹

Pwning|SH1TTY (finals)

- 2 idiots, 1 keyboard
- Linux kernel module, keylogger
- Qemu + ramfs
- 2 modes:
  - DUMB: log *
  - «SMART»: log passwords

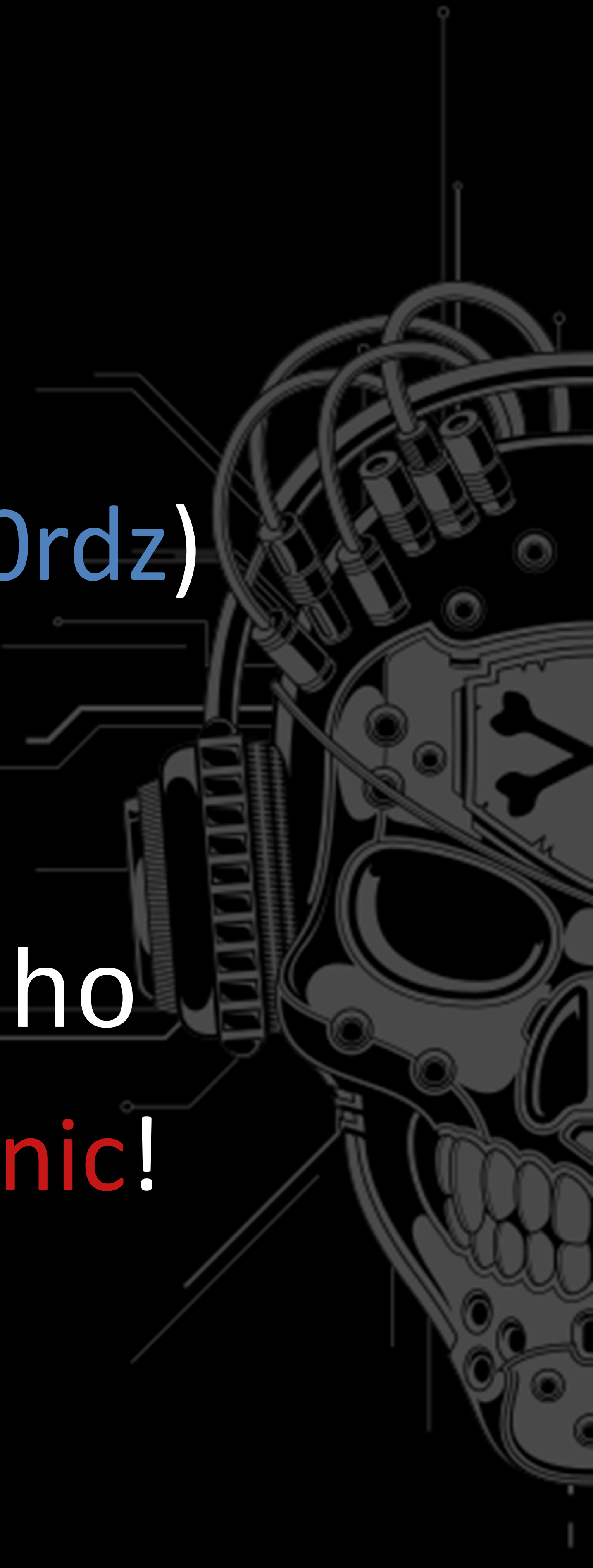
https://www.youtube.com/watch?v=u8qgehH3kEQ

Pwning|SH1TTY (finals)

- TTY keylogger:
  - Creates a line discipline based on N_TTY
  - Change `ldisc.ops->receive_buf2`
  - Replace N_TTY with our ldisc
- The TTY demystified: http://www.linusakesson.net/programming/tty/

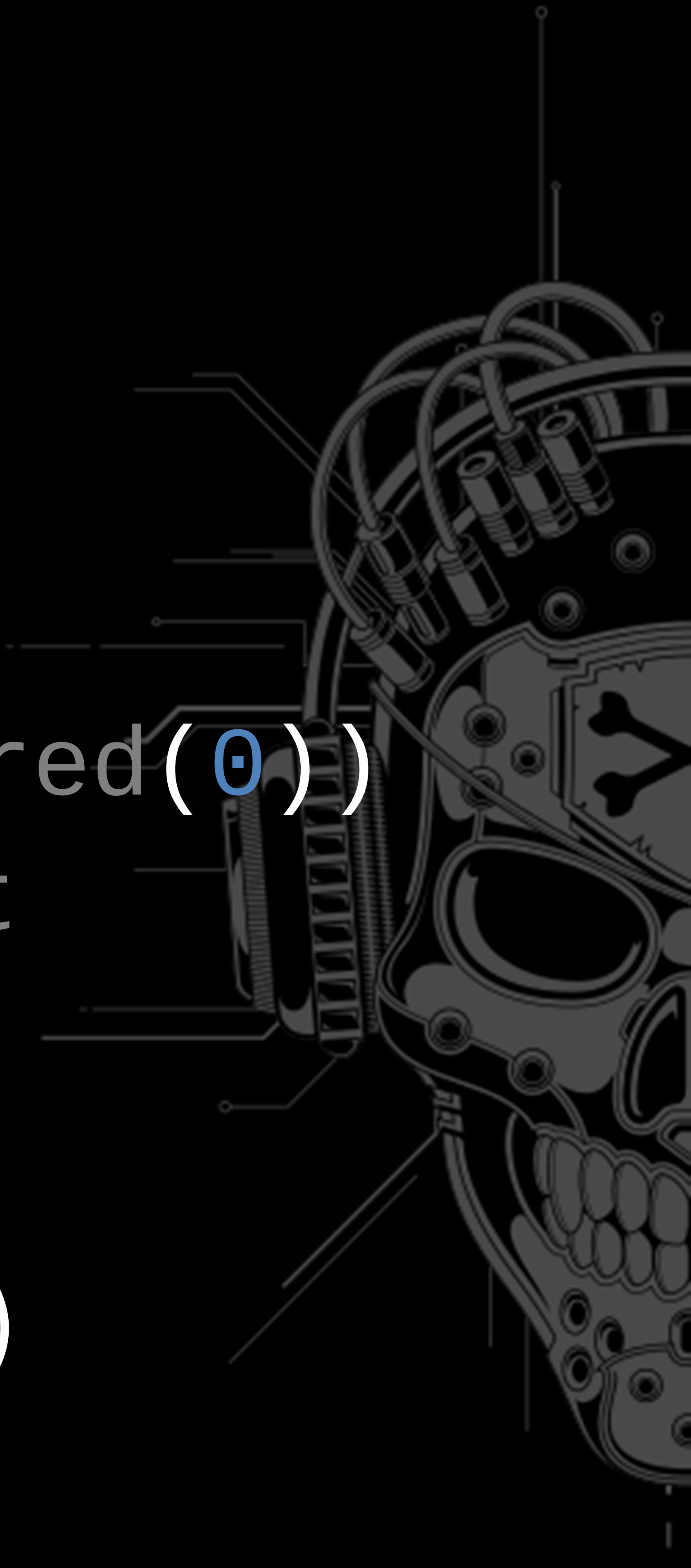
Pwning|SH1TTY (finals)

- Vulnerability:
  - Go to SMART mode (type G1v3m3p4ssw0rdz)
  - A user enters a password if
    - `L_ICANON(tty) && !L_ECHO(tty)`
  - Change the line settings with `stty -echo`
  - Type a very long password ⇨💥kernel panic!



Pwning|SH1TTY (finals)

- Classic kstack buffer overflow
- Upload a binary
  - Create a function that does the classic `commit_creds(prepare_kernel_cred(0))`
  - End the function with a `swapgs ; iret`
  - No SMEP/KERNEXEC
- Unless… you can't?
  - Also, running in another context (*kworker*)



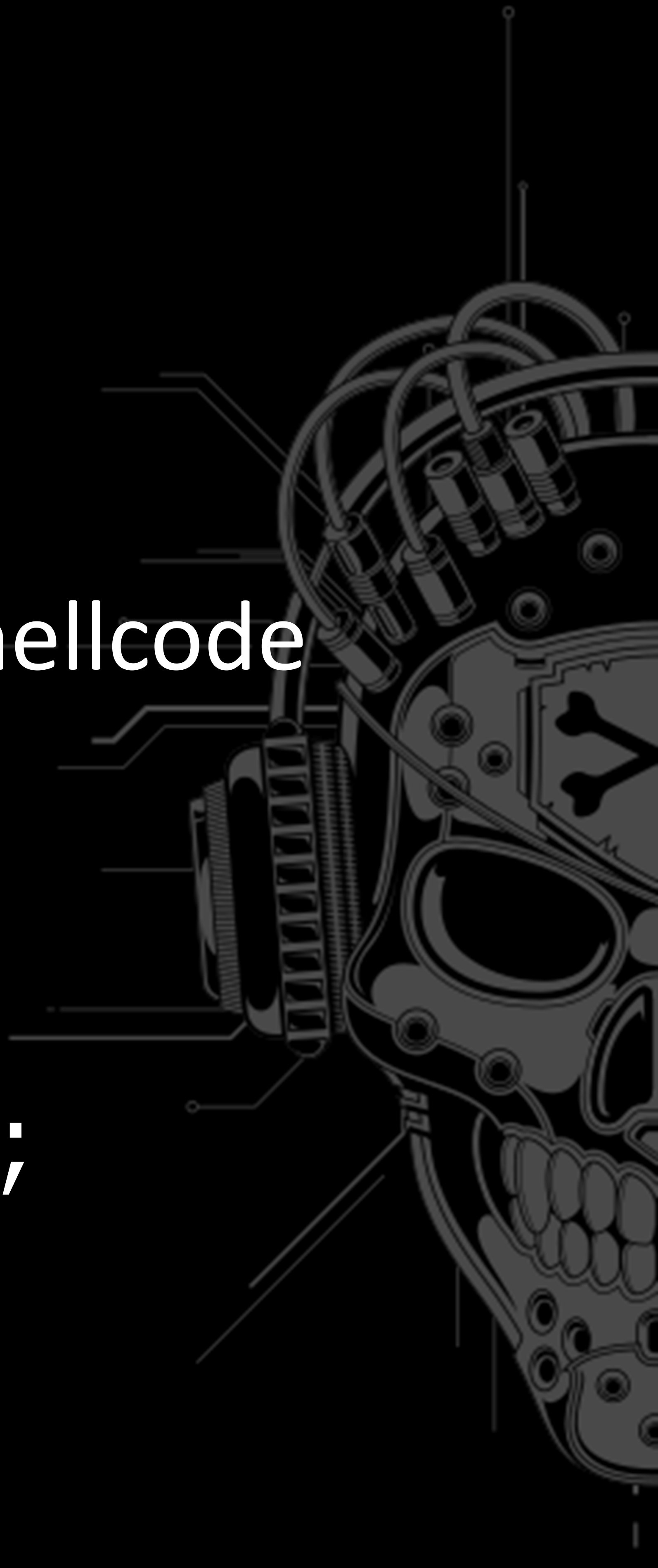
Pwning|SH1TTY (finals)

- Write the exploit «*with your bare hands*»

Pwning|SH1TTY (finals)

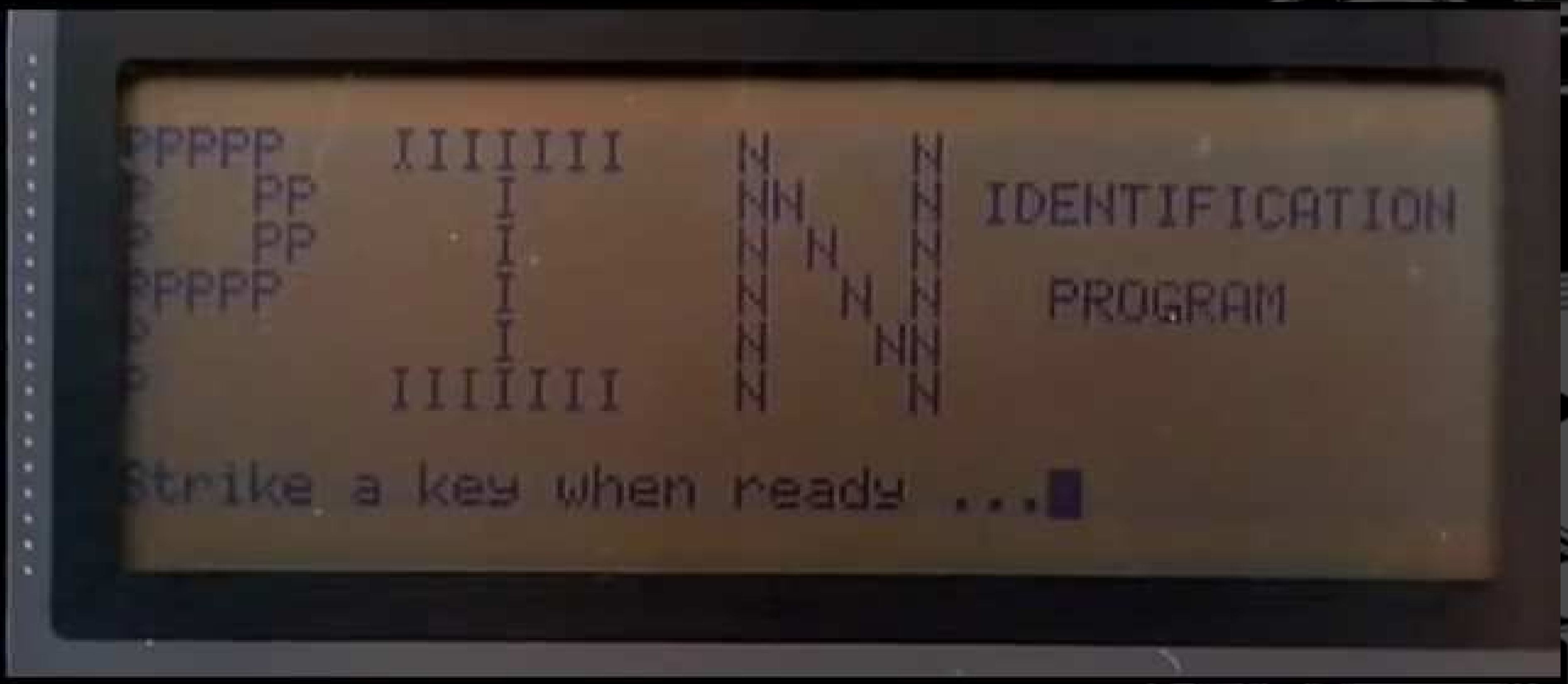
- Must use ROP (x64 kstack is NX)
  - Full ROP
  - Create/reuse RWX section ; copy/jmp to shellcode
- Payload:

```
pid = find_get_pid(shell_pid);
task = get_pid_task(pid);
creds = prepare_kernel_cred(0);
task->cred = creds;
```



Reversing100 | Baby Haskell

- Haskell binary
- Lazy evaluation
- Timer to prevent debugging
- Impossible to reverse statically



IDENTIFICATION PROGRAM

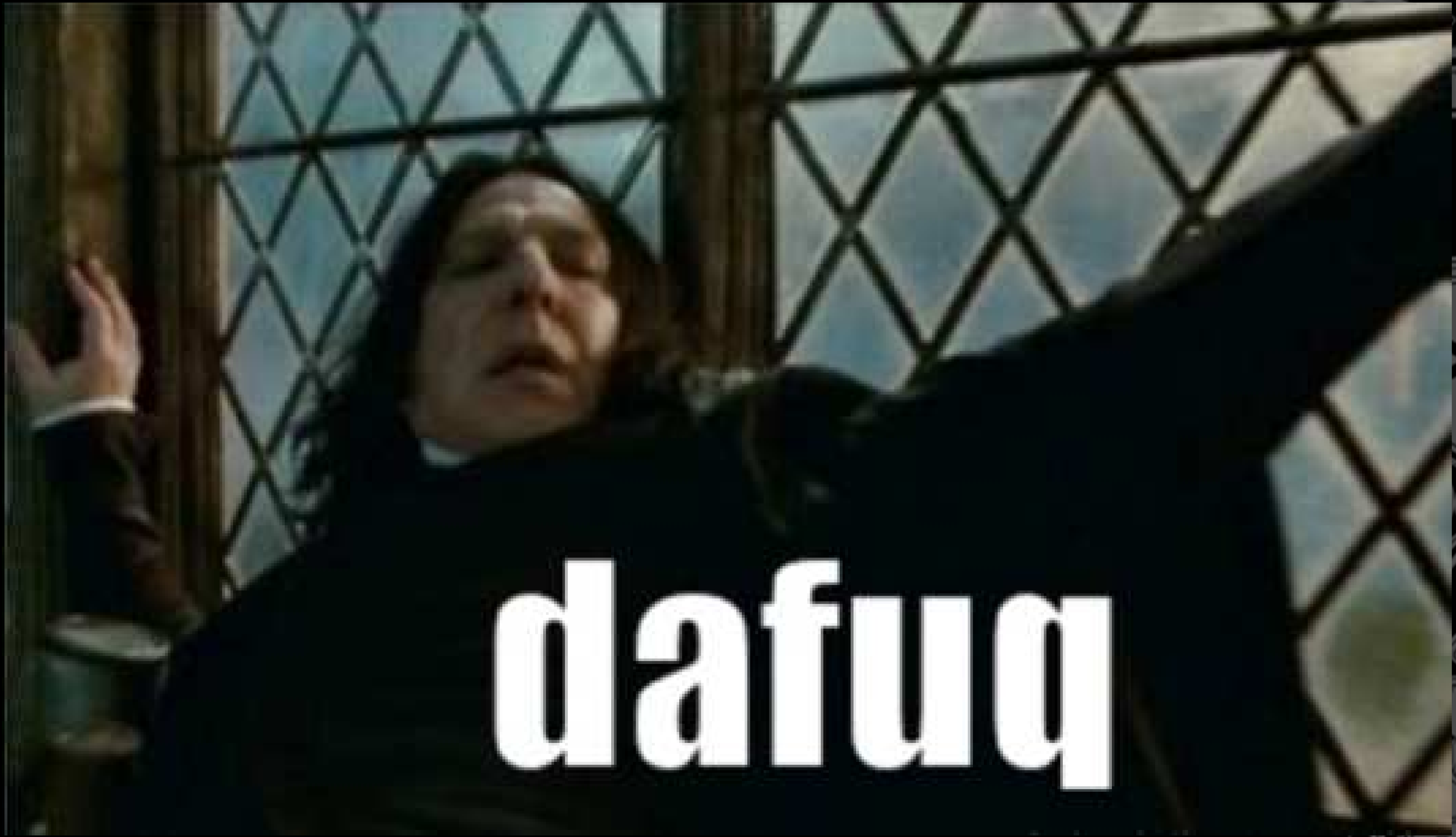
Strike a key when ready ...

https://www.youtube.com/watch?v=AqtMOUb3g6g

Reversing100 | Baby Haskell

- Impossible?

- Well go see in IDA for yourself

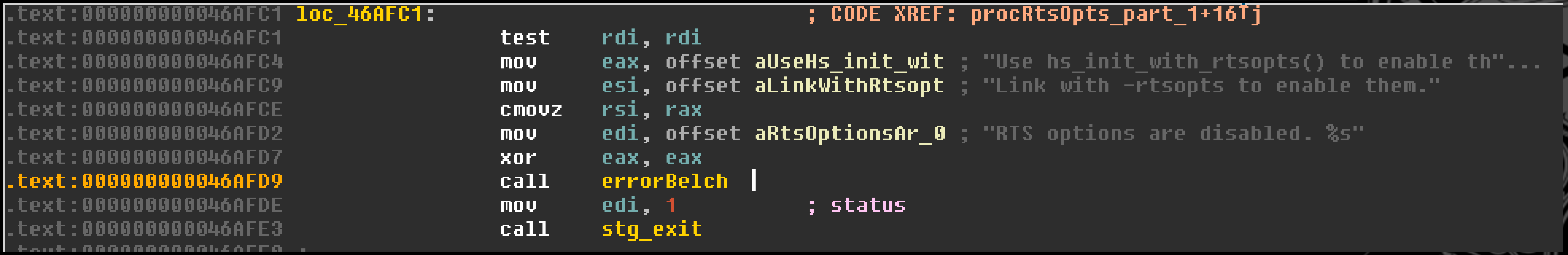


Reversing100 | Baby Haskell

- RTFM!
- Haskell Runtime System (RTS)
- Options to profile the binary when running it
- See allocated memory, threads, etc.
- Pass options +RTS in args
- … if compiled `--with-rtsopts`

Reversing100|Baby Haskell

- RTS options are disabled. Link with -rtsopts to enable them.



```
.text:000000000046AFC1 loc_46AFC1:                             ; CODE XREF: procRtsOpts_part_1+16↑j
.text:000000000046AFC1                 test    rdi, rdi
.text:000000000046AFC4                 mov     eax, offset aUseHs_init_wit ; "Use hs_init_with_rtsopts() to enable th"...
.text:000000000046AFC9                 mov     esi, offset aLinkWithRtsopt ; "Link with -rtsopts to enable them."
.text:000000000046AFCE                 cmovz   rsi, rax
.text:000000000046AFD2                 mov     edi, offset aRtsOptionsAr_0 ; "RTS options are disabled. %s"
.text:000000000046AFD7                 xor     eax, eax
.text:000000000046AFD9                 call    errorBelch
.text:000000000046AFDE                 mov     edi, 1          ; status
.text:000000000046AFE3                 call    stg_exit
```

- Patch the binary in the xref (jz to jnz ☺)

Reversing100|Baby Haskell

```
$ ./haskell.bin +RTS -t --machine-readable -RTS INS{a
Nope
 [("bytes allocated", "54520")
 ,("num_GCs", "1")
 ,("average_bytes_used", "44312")
 ,("max_bytes_used", "44312")
 ,("num_byte_usage_samples", "1")
 ,("peak_megabytes_allocated", "1")
 ,("init_cpu_seconds", "0.00")
 ,("init_wall_seconds", "0.00")
 ,("mutator_cpu_seconds", "0.00")
 ,("mutator_wall_seconds", "0.00")
 ,("GC_cpu_seconds", "0.00")
 ,("GC_wall_seconds", "0.00")
 ]
```

Reversing100|Baby Haskell

```
$ ./haskell.bin +RTS -t --machine-readable -RTS INS{Y
Nope
 [("bytes allocated", "54592")
 ,("num_GCs", "1")
 ,("average_bytes_used", "44312")
 ,("max_bytes_used", "44312")
 ,("num_byte_usage_samples", "1")
 ,("peak_megabytes_allocated", "1")
 ,("init_cpu_seconds", "0.00")
 ,("init_wall_seconds", "0.00")
 ,("mutator_cpu_seconds", "0.00")
 ,("mutator_wall_seconds", "0.00")
 ,("GC_cpu_seconds", "0.00")
 ,("GC_wall_seconds", "0.00")
 ]
```

Reversing100|Baby Haskell

```
$ ./haskell.bin +RTS -t --machine-readable –RTS \
INS{You_5h0uld_1earn_HASKELL}
Congratz
 [("bytes allocated", "61528")
 ,("num_GCs", "1")
 ,("average_bytes_used", "44312")
 ,("max_bytes_used", "44312")
 ,("num_byte_usage_samples", "1")
 ,("peak_megabytes_allocated", "1")
 ,("init_cpu_seconds", "0.00")
 ,("init_wall_seconds", "0.00")
 ,("mutator_cpu_seconds", "0.00")
 ,("mutator_wall_seconds", "0.00")
 ,("GC_cpu_seconds", "0.00")
 ,("GC_wall_seconds", "0.00")
 ]
```

INSOMNI'HACK

Web100|YNOS

- WebRPC
- Leak the sources

https://www.youtube.com/watch?v=O2rGTXHvPCQ

- More than one way to solve it

Web100|YNOS

- SQLi in login
  Read source
- Deserialization
  of json data
- Code execution by using ReflectionFunction

Issues|YNOS

- APParmor
  Prevent FILE
  access in
  MYSQL
- People still
  Managed to
  Solve it before we eventually fixed it

Pwn100|Elysium

- Also known as Esylium…
- 32 bits; Partial RELRO, No canary, NX, PIE
- AES-CBC encrypted protocol
- Commands format is
  - `<sha1(cmd)>:<cmd>`
- Manipulate the number of each Elysium units (*Medical*, *Military*, *Social*, *Spy*…)
- Units are stored in global variables, aka `.data` section

Pwn100 | Elysium

- Vulnerabilities:
  - Possible to add negative units
  - Path traversal in `get_informations`
    - Easy PIE bypass: leak `/proc/self/maps`
    - Cannot read the flag directly
  - `sscanf(input, "%[^:]:%[^\n]", &sha1, cmd);`

Pwn100|Elysium

- Straight-forward ROP then?
- `free(ptr1); free(ptr2);`
- Randomized Heap layout
- How to survive a `free`?
  - Point to a valid heap chunk
  - `free(NULL)`

Pwn100 | Elysium

- Exploit:
  - Leak `/proc/self/maps` to bypass PIE
  - Use units to craft fake heap chunks in .data
  - Overwrite ptr1 and ptr2 with the .data chunks
  - Use a read/mprotect/jmp shellcode ropchain
  - Launch the setuid «`reboot`» binary to get the flag

Pwn300|ARPS-331

- Pcap + URL
- Custom HTTP methods
- Lame web interface

https://www.youtube.com/watch?v=2efhrCxI4J0

Pwn300 | ARPS-331

- Playlist
- Add movies
  - Local
  - Remote
- PCAP infos
  ⇨ Rickroll

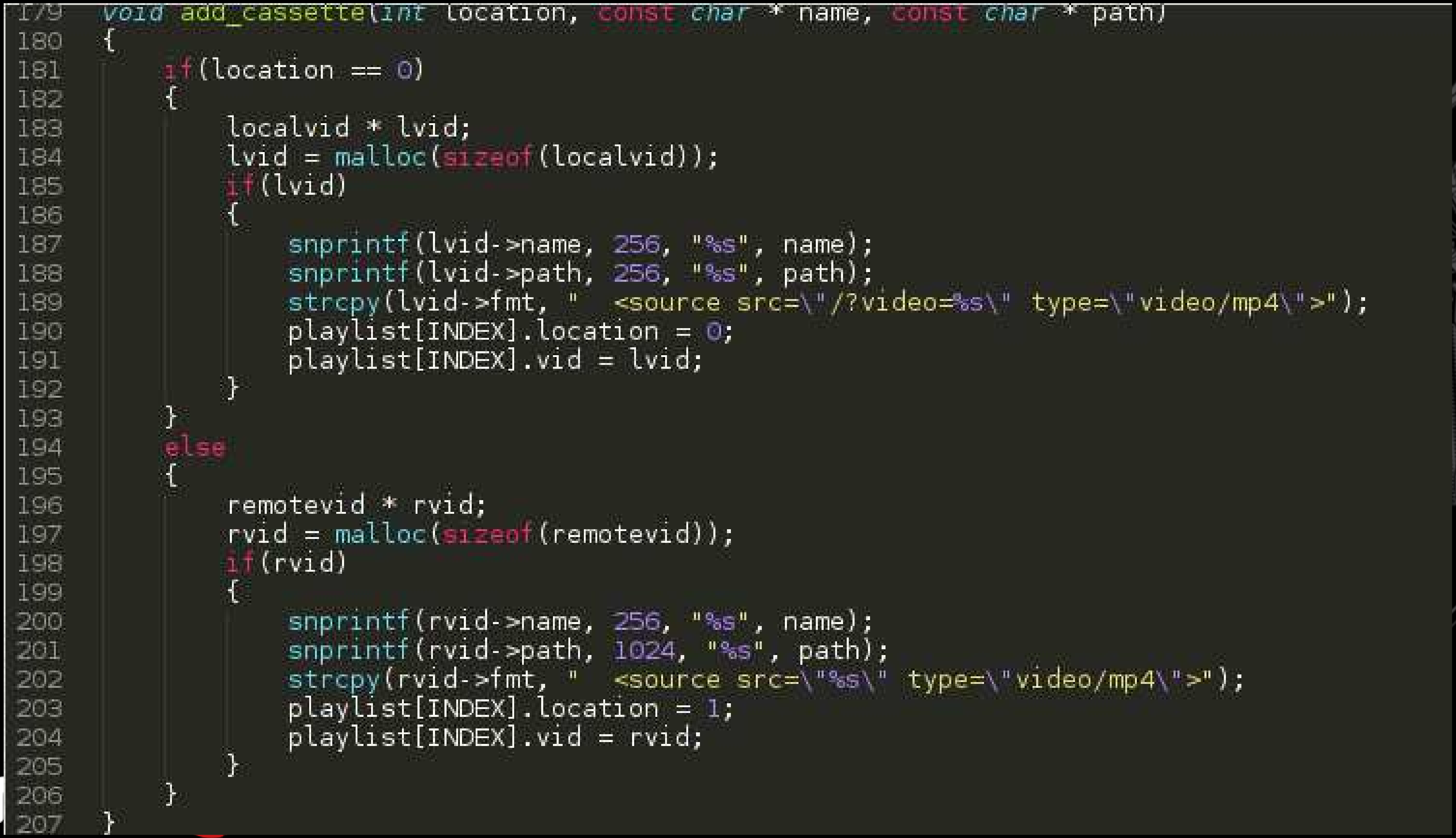
Pwn300|ARPS-331

- LFI, eg `/proc/self/maps`
- Get module file, libc, apache config, etc.
- Type confusion local/remote remote ⇨ bigger

```
typedef struct {
        char name[256];
        char path[256];
        char fmt[128];
} localvid;

typedef struct {
        char name[256];
        char path[1024];
        char fmt[128];
} remotevid ;
```

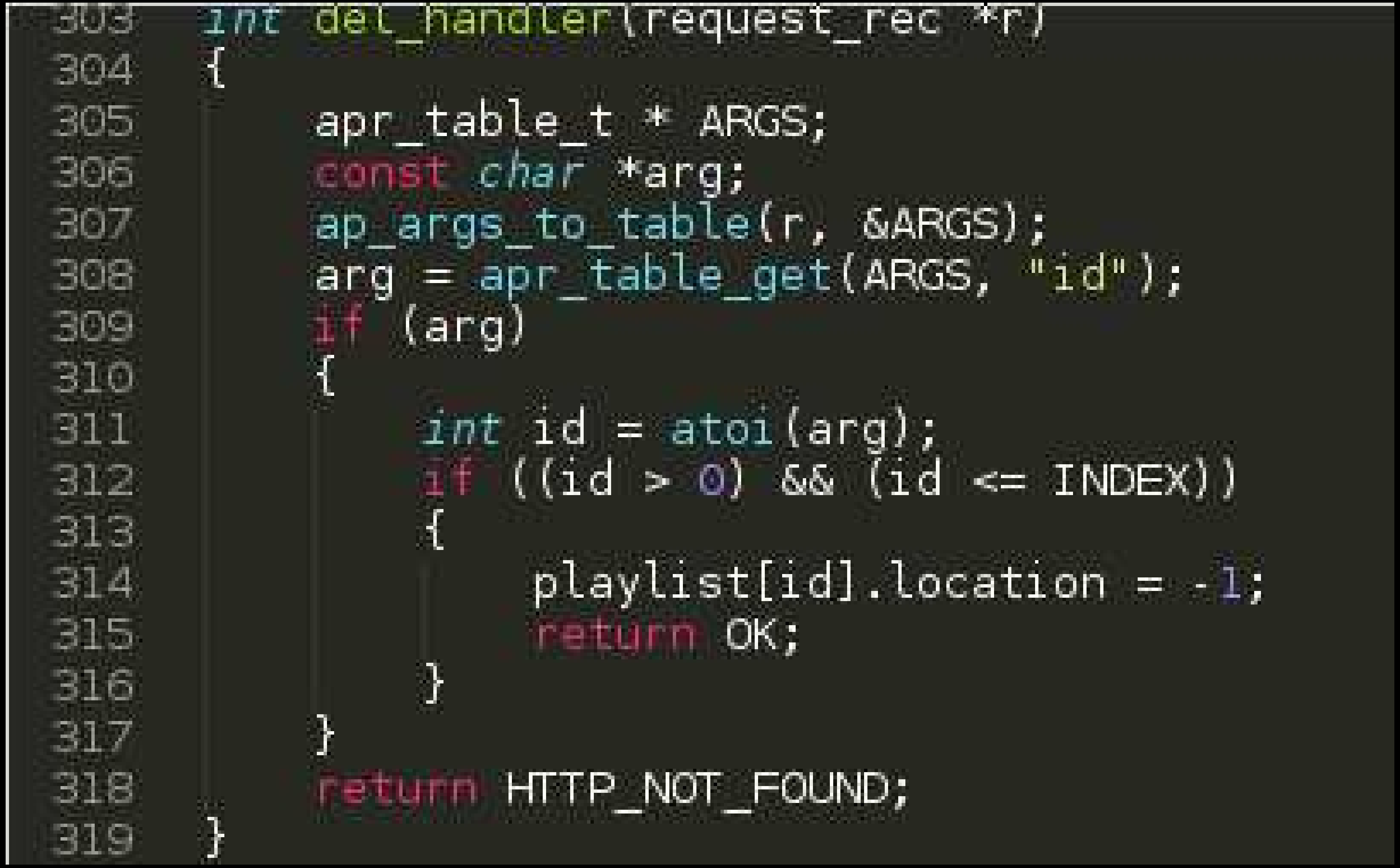
Pwn300|ARPS-331



```
void add_cassette(int location, const char * name, const char * path)
{
    if(location == 0)
    {
        localvid * lvid;
        lvid = malloc(sizeof(localvid));
        if(lvid)
        {
            snprintf(lvid->name, 256, "%s", name);
            snprintf(lvid->path, 256, "%s", path);
            strcpy(lvid->fmt, "  <source src=\"/?video=%s\" type=\"video/mp4\">");
            playlist[INDEX].location = 0;
            playlist[INDEX].vid = lvid;
        }
    }
    else
    {
        remotevid * rvid;
        rvid = malloc(sizeof(remotevid));
        if(rvid)
        {
            snprintf(rvid->name, 256, "%s", name);
            snprintf(rvid->path, 1024, "%s", path);
            strcpy(rvid->fmt, "  <source src=\"%s\" type=\"video/mp4\">");
            playlist[INDEX].location = 1;
            playlist[INDEX].vid = rvid;
        }
    }
}
```

Pwn300|ARPS-331

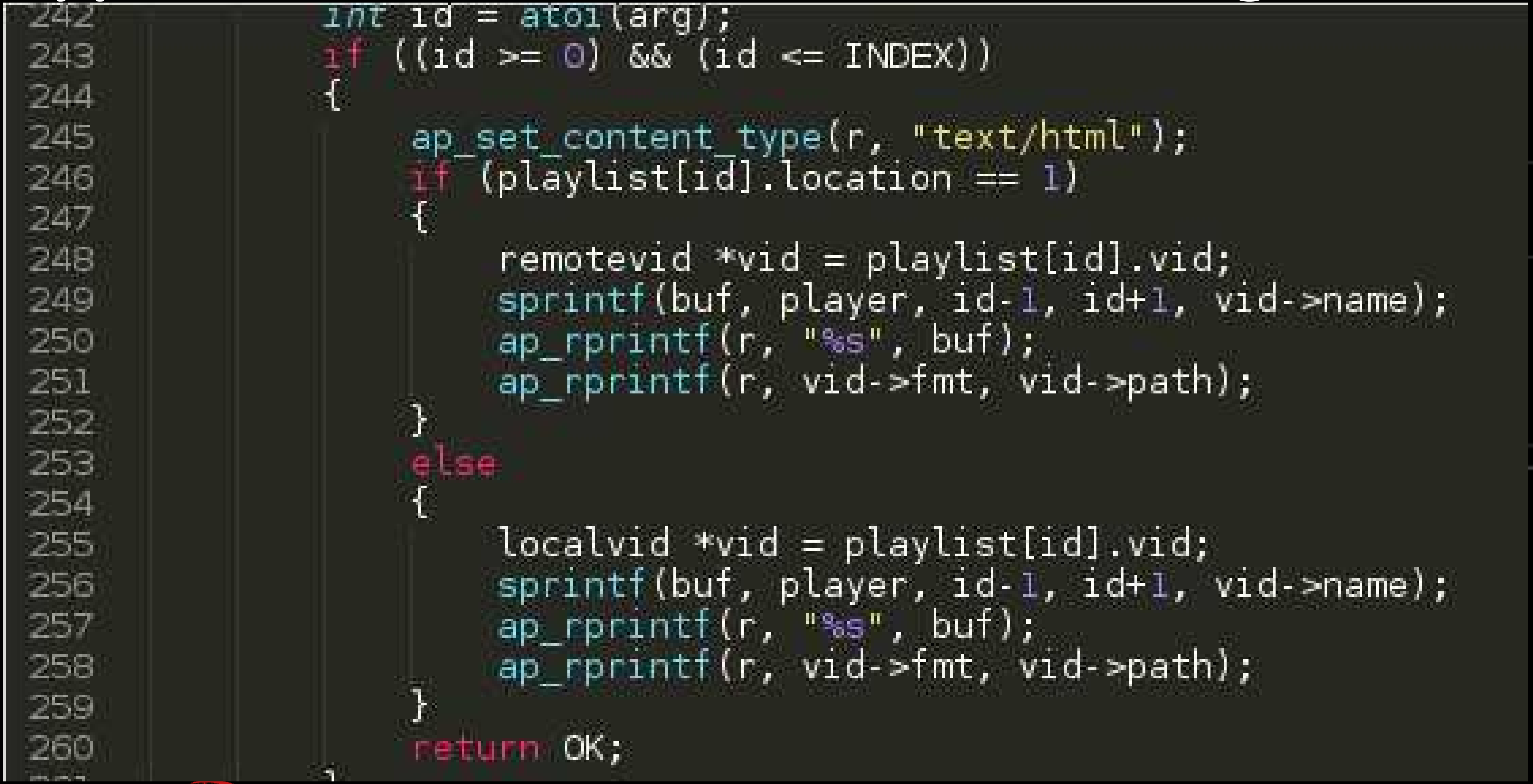
- Confuse the type



```
int del_handler(request_rec *r)
{
    apr_table_t * ARGS;
    const char *arg;
    ap_args_to_table(r, &ARGS);
    arg = apr_table_get(ARGS, "id");
    if (arg)
    {
        int id = atoi(arg);
        if ((id > 0) && (id <= INDEX))
        {
            playlist[id].location = -1;
            return OK;
        }
    }
    return HTTP_NOT_FOUND;
}
```

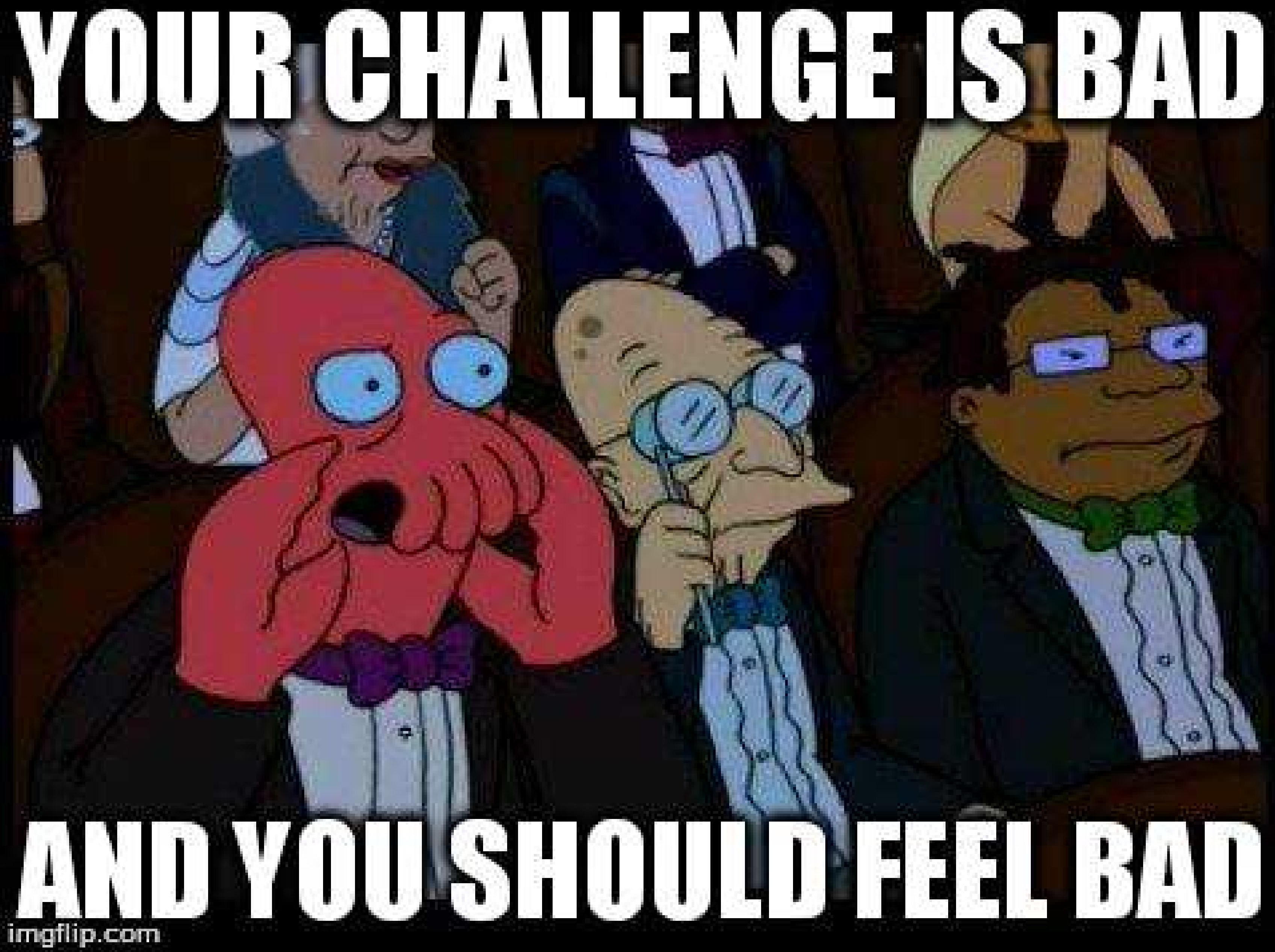
Pwn300|ARPS-331

- Type confusion => Format string



```
        int id = atoi(arg);
        if ((id >= 0) && (id <= INDEX))
        {
            ap_set_content_type(r, "text/html");
            if (playlist[id].location == 1)
            {
                remotevid *vid = playlist[id].vid;
                sprintf(buf, player, id-1, id+1, vid->name);
                ap_rprintf(r, "%s", buf);
                ap_rprintf(r, vid->fmt, vid->path);
            }
            else
            {
                localvid *vid = playlist[id].vid;
                sprintf(buf, player, id-1, id+1, vid->name);
                ap_rprintf(r, "%s", buf);
                ap_rprintf(r, vid->fmt, vid->path);
            }
            return OK;
        }
```

Pwn300 | ARPS-331



Pwn300|ARPS-331

- FORTIFY_SOURCE=2
- But…

- `ap_*intf` reimplements all formats
- … including `%n`

Pwn300|ARPS-331

- Debug Apache with eg one byte int3
- Analyze stack when FMT
- Overwrite SEIP with your ROP chain

```
_libc_system
0xdeadbeef
"/bin/sh"
```

Pwn300 | ARPS-331

- Create a pointer to SEIP on the stack
- Dereference the pointer %hn
- Double encode format string
- Increment pointer
- Etc.
- HTTP keep-alive

Issues|ARPS-331

- Apache module documentation
  - Custom methods howto ?
    ⇨ no results on stackoverflow ☹
- Originally 64bit
  - NULL bytes ? Not gonna work…

Issues|ARPS-331

- Multiple sessions in same process
  - Apache mpm-itk FTW
  - Double fork ⇨ isolate users per process
- Finished and validated 6h before start
  - Still plenty of time ☺

Pwn300 | interview

- POP3 server
- Hack into sony-mailserver in less than 60 seconds
- Credentials provided to each team, along with the SMTP server
- x64, Full RELRO, SSP, NX, PIE, FORTIFY_SOURCE

Pwn300 | interview

- POP3 101:
  - Simple mail retrieving service
  - Cannot open 2 sessions for a user
  - APOP, DELE, LIST, NOOP, PASS, QUIT, RETR, RSET, STAT, TOP, UIDL, USER
  - Guessed the vulnerability already?

Pwn300|interview

- TOP :
  - read first n lines of an email
  - Internally using `realloc()`
- DELE :
  - Deletes an email
  - Doesn't actually remove the file until QUIT
  - Internally `free()` the `top_block->text` and `top_block`
- RSET :
  - Cancels any previous operation, unDELEtes files

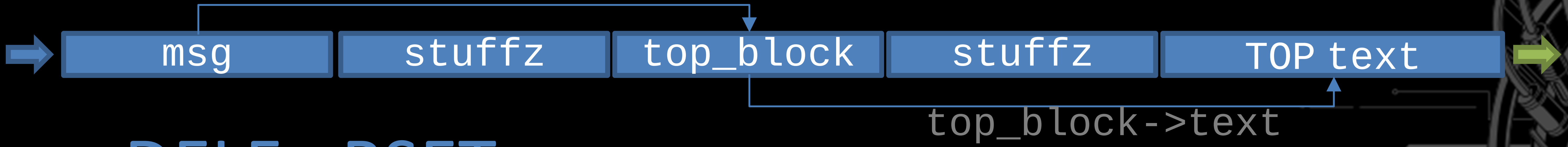
```
typedef struct top_block {
           size_t size;
           size_t linecount;
           char *text;
           size_t header_size;
} top_block_t;
```

Pwn300|interview

- BUG: missing `msg->top_block = NULL;`
- Use-After-Free
  - TOP to allocate a `top_block`
  - Free it using DELE
  - RSET to make it usable again
  - Manage to allocate data to overwrite the free chunk
  - TOP again to append data ⇨ Write-What-Where

Pwn300 | interview

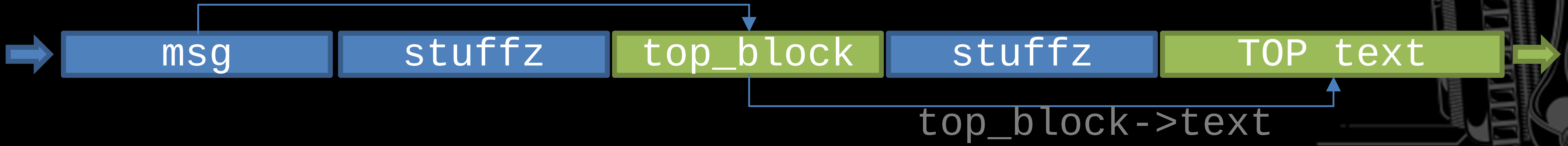
- TOP 0



msg | stuffz | top_block | stuffz | TOP text

top_block->text

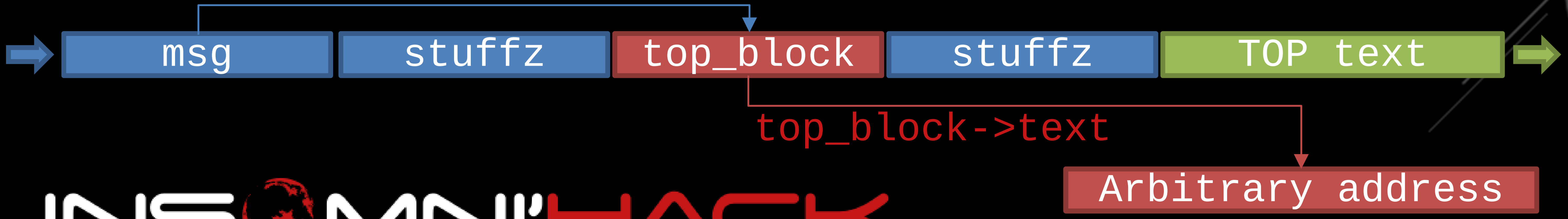
- DELE ; RSET



msg | stuffz | top_block | stuffz | TOP text

top_block->text

- TOP 1



msg | stuffz | top_block | stuffz | TOP text

top_block->text

Arbitrary address

Pwn300|interview

- Exploit part 1: leak pointers
  - Send 2 messages
  - TOP each message
  - DELE each message
  - RSET
  - TOP message 1 ⇨ leak a heap pointer
  - TOP message 2 ⇨ leak a libc pointer
  - Empty the mailbox for next part

Pwn300|interview

- Exploit part 2:
  - Craft a fake `top_block` in a message line of ~ `sizeof(struct top_block)`
  - Send that message a few times
  - TOP then DELE the first half of messages
  - TOP the second part to fill `free()`'d chunks
  - TOP last message of the first half
  - Write-What-Where primitive

Pwn300 | interview

- Write-What, Where?
  - Full RELRO!
  - The hard way: overwrite the stack
    - Overwrite a file name in the heap with `/proc/self/maps`, then RETR it
    - Leak a portion of the stack, find offset to a saved-RIP
    - Write a ropchain at this offset

Pwn300 | interview

- Write-What, Where?
  - Easier: overwrite a libc pointer
    - Libc has many pointers that you can target
    - Not always easy to pivot to a ropchain
    - In this challenge: __free_hook
    - Overwrite with &system
    - free("/bin/bash <&4 >&4")

Pwn300 | interview

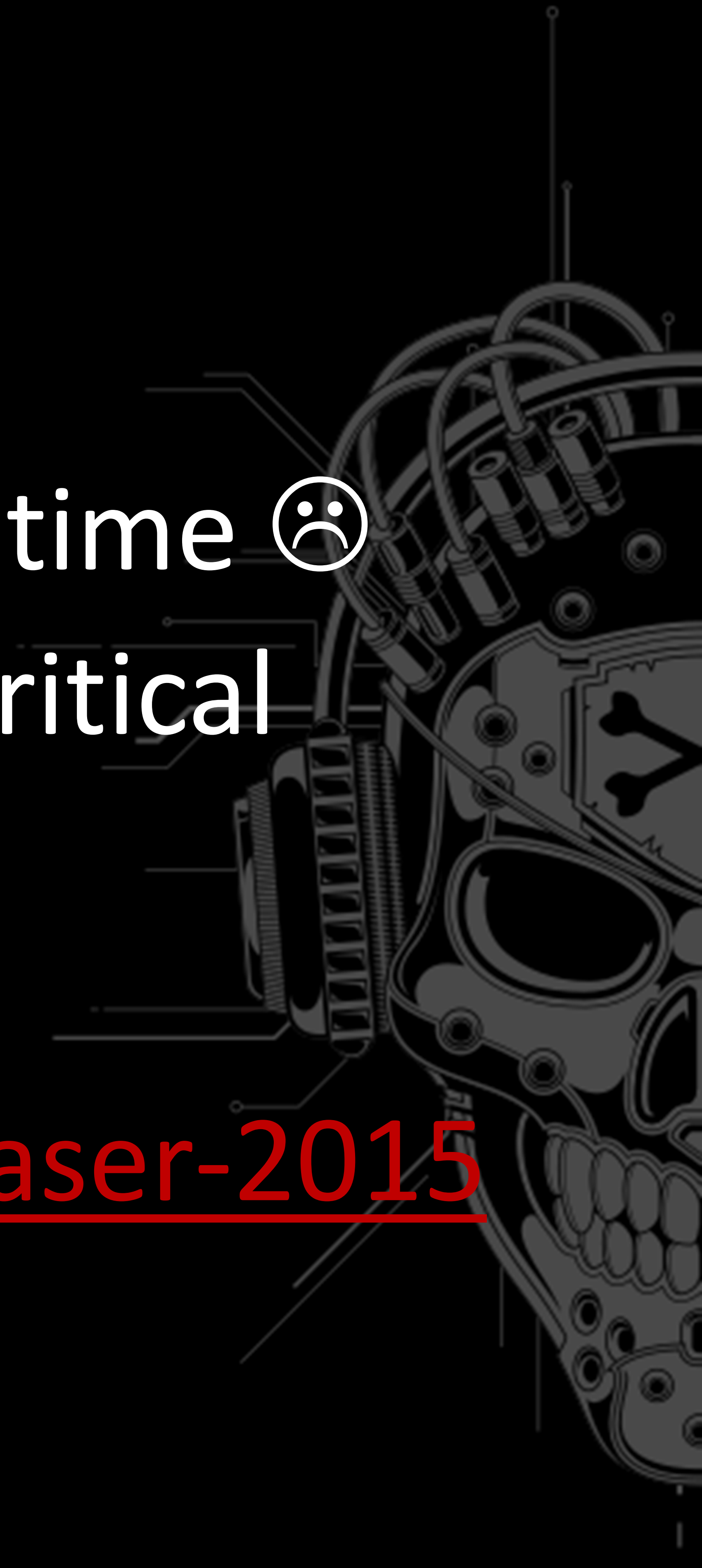
- Putting the pieces together

https://www.youtube.com/watch?v=u1Ds9CeG-VY

Conclusions

- Missing some easier challs
- Another CTF took place at the same time ☹
- Few issues during the CTF, nothing critical

- Sources: https://github.com/Insomnihack/Teaser-2015



Conclusions|Questions/Contact

- Questions ?

- Twitter:
  - @0xGrimmlin
  - @__awe