Workshop: Intro to CTF

What is a CTF?

- Capture The Flag
- Compete as a team or individually
- Exploit vulnerabilities to collect “flags”
    - TLDR; legal hacking
    - Flag example: “TD{this-is-a-flag}”
- Flags give X amounts of points
- Most points collected = 🏆

Web Exploitation

Exploiting web pages

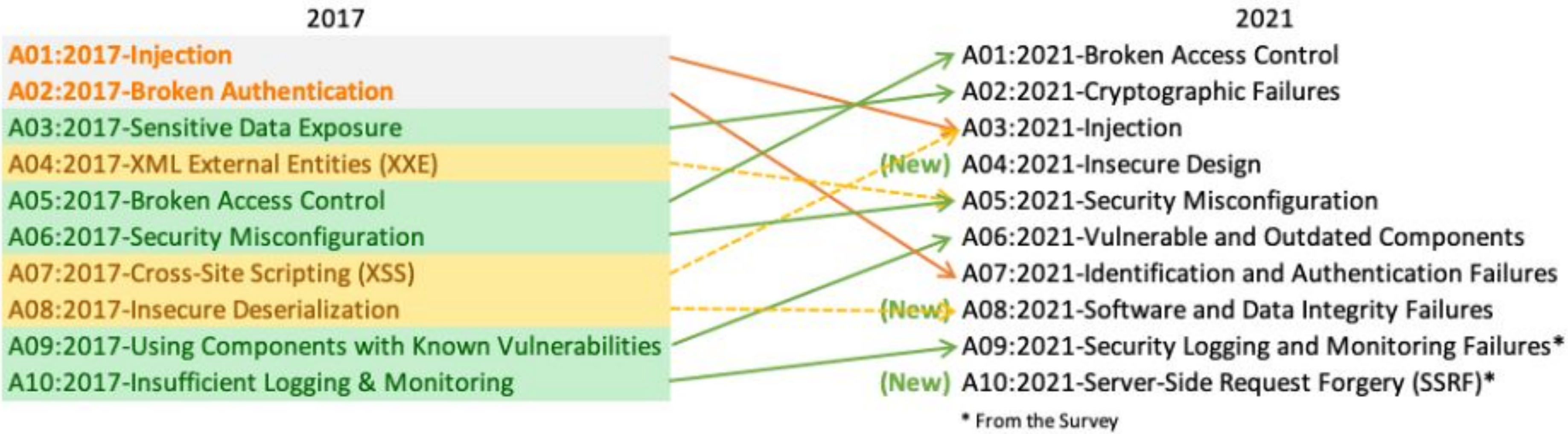
- Various programming languages*
- Issues fundamental to the internet
- Misconfiguration

Examples

- SQL-Injections
- Command-Injections
- Directory Traversal
- Cross Site Scripting (XSS)
- Cross Site Request Forgery (CSRF)

*Javascript, PHP, Python etc

OWASP Top 10



2017

- A01:2017-Injection
- A02:2017-Broken Authentication
- A03:2017-Sensitive Data Exposure
- A04:2017-XML External Entities (XXE)
- A05:2017-Broken Access Control
- A06:2017-Security Misconfiguration
- A07:2017-Cross-Site Scripting (XSS)
- A08:2017-Insecure Deserialization
- A09:2017-Using Components with Known Vulnerabilities
- A10:2017-Insufficient Logging & Monitoring

2021

- A01:2021-Broken Access Control
- A02:2021-Cryptographic Failures
- A03:2021-Injection
- (New) A04:2021-Insecure Design
- A05:2021-Security Misconfiguration
- A06:2021-Vulnerable and Outdated Components
- A07:2021-Identification and Authentication Failures
- (New) A08:2021-Software and Data Integrity Failures
- A09:2021-Security Logging and Monitoring Failures*
- (New) A10:2021-Server-Side Request Forgery (SSRF)*

* From the Survey

https://owasp.org/www-project-top-ten/

Crypto

- Crack encryption to access encrypted content, e.g.
    - brute force search of possible keys
    - partially known key or plaintext
    - exploit usage faulty or unsafe crypto mechanisms

- Useful tools
    - dcode.fr: https://www.dcode.fr/
    - Cyberchef: https://gchq.github.io/CyberChef/

Crypto - XOR

- Symmetric encryption, one key
- One-time pad is uncrackable (generally)
- Message ⊕ Key = Encrypted
- Encrypted ⊕ Key = Message
- Encrypted ⊕ Message = Key

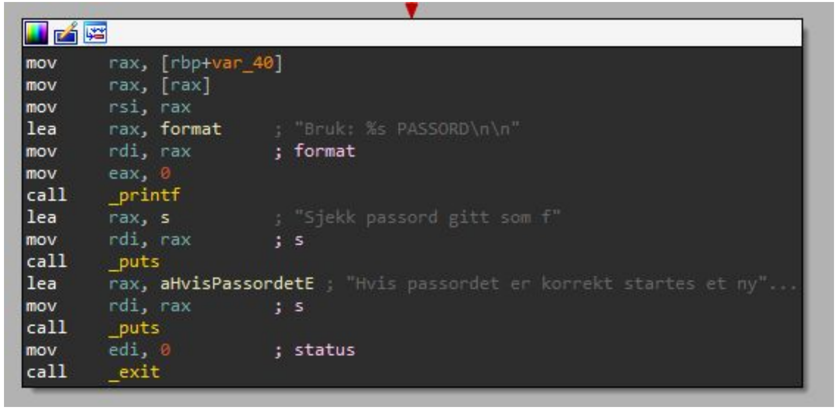
Crypto - RSA

- Asymmetric encryption, two keys
  - Secret key: two primes **P** and **Q**
  - Public key: two numbers **N** and **E**
- Can encrypt messages by only knowing **N** and **E**
  - Enc = Msg^E mod N
- Cannot decrypt the messages without also knowing **D**
  - phi = (p-1)*(q-1)
  - D = E^-1 mod phi
  - Msg = Enc^D mod N
- "*Safe*" because brute force search after **P** and **Q** is computationally expensive

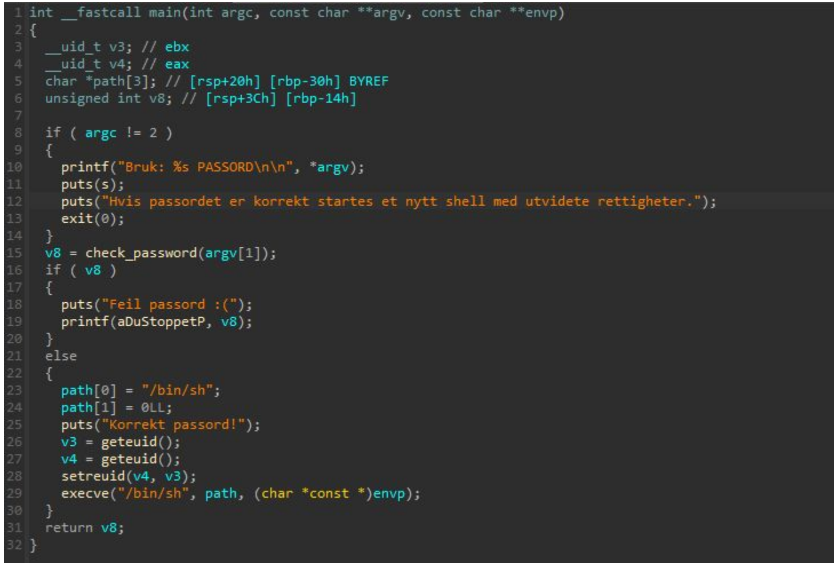
Reversing

- Reverse engineer compiled programs
  - Find out what the program does
  - e.g. Malware

- Useful tools
  - IDA (https://hex-rays.com/ida-free/)
  - Ghidra (https://ghidra-sre.org/)



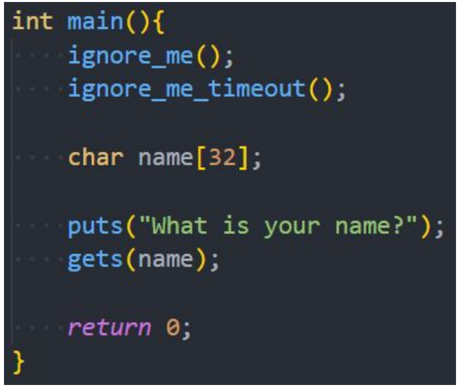
```
mov     rax, [rbp+var_40]
mov     rax, [rax]
mov     rsi, rax
lea     rax, format      ; "Bruk: %s PASSORD\n\n"
mov     rdi, rax         ; format
mov     eax, 0
call    _printf
lea     rax, s           ; "Sjekk passord gitt som f"
mov     rdi, rax         ; s
call    _puts
lea     rax, aHvisPassordetE ; "Hvis passordet er korrekt startes et ny"...
mov     rdi, rax         ; s
call    _puts
mov     edi, 0           ; status
call    _exit
```



```
int __fastcall main(int argc, const char **argv, const char **envp)
{
  __uid_t v3; // ebx
  __uid_t v4; // eax
  char *path[3]; // [rsp+20h] [rbp-30h] BYREF
  unsigned int v8; // [rsp+3Ch] [rbp-14h]

  if ( argc != 2 )
  {
    printf("Bruk: %s PASSORD\n\n", *argv);
    puts(s);
    puts("Hvis passordet er korrekt startes et nytt shell med utvidete rettigheter.");
    exit(0);
  }
  v8 = check_password(argv[1]);
  if ( v8 )
  {
    puts("Feil passord :(");
    printf(aDuStoppetP, v8);
  }
  else
  {
    path[0] = "/bin/sh";
    path[1] = 0LL;
    puts("Korrekt passord!");
    v3 = geteuid();
    v4 = geteuid();
    setreuid(v4, v3);
    execve("/bin/sh", path, (char *const *)envp);
  }
  return v8;
}
```

Pwn



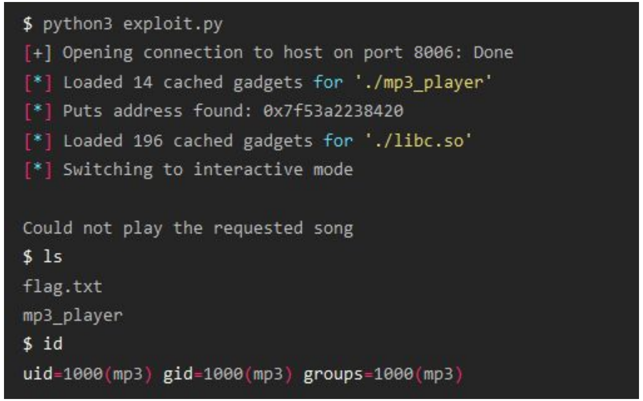
```c
int main(){
    ignore_me();
    ignore_me_timeout();

    char name[32];

    puts("What is your name?");
    gets(name);

    return 0;
}
```

- Usually **Binary Exploitation** in CTFs
  - e.g. by being given a C program
- Make the program behave *unintended*
- Useful tools
  - Python library: Pwntools (https://github.com/Gallopsled/pwntools)
  - Debugging: Pwndbg / gef (https://github.com/pwndbg/pwndbg / https://github.com/hugsy/gef)



```
$ python3 exploit.py
[+] Opening connection to host on port 8006: Done
[*] Loaded 14 cached gadgets for './mp3_player'
[*] Puts address found: 0x7f53a2238420
[*] Loaded 196 cached gadgets for './libc.so'
[*] Switching to interactive mode

Could not play the requested song
$ ls
flag.txt
mp3_player
$ id
uid=1000(mp3) gid=1000(mp3) groups=1000(mp3)
```

Miscellaneous

- If you can’t label it… then misc.
- Puzzles and games, e.g.
    - “People speak in my name. I am written and read and often executed, but can do neither myself. What am I?”
    - very hard flappy bird game
- IRL/on-site challenges and activities, e.g.
    - rebus
- Steganography, Machine Learning, etc.

Where do I find CTFs?

- CTFtime
- PicoCTF
- OverTheWire
- CryptoHack
- Pwn College
- HackTheBox
- **UiTHack**