Ethical Student Hackers

How to Play a CTF

The Legal Bit

- The skills taught in these sessions allow identification and exploitation of security vulnerabilities in systems. We strive to give you a place to practice legally, and can point you to other places to practice. These skills should not be used on systems where you do not have explicit permission from the owner of the system. It is VERY easy to end up in breach of relevant laws, and we can accept no responsibility for anything you do with the skills learnt here.

- If we have reason to believe that you are utilising these skills against systems where you are not authorised you will be banned from our events, and if necessary the relevant authorities will be alerted.

- Remember, if you have any doubts as to if something is legal or authorised, just don't do it until you are able to confirm you are allowed to.

- Relevant UK Law: https://www.legislation.gov.uk/ukpga/1990/18/contents

Code of Conduct

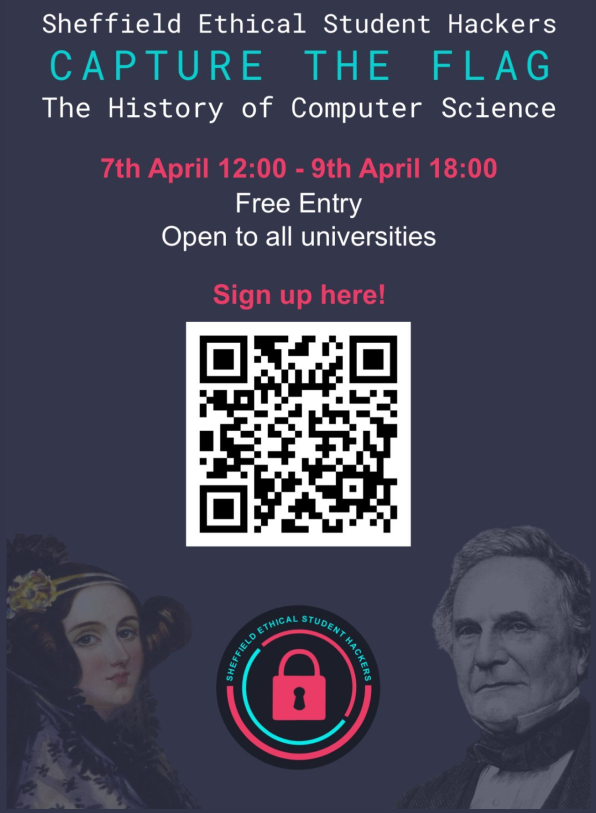
- Before proceeding past this point you must read and agree to our Code of Conduct - this is a requirement from the University for us to operate as a society.

- If you have any doubts or need anything clarified, please ask a member of the committee.

- Breaching the Code of Conduct = immediate ejection and further consequences.

- Code of Conduct can be found at https://shefesh.com/downloads/SESH%20Code%20of%20Conduct.pdf

SESH CTF SIGNUP!



Sheffield Ethical Student Hackers

CAPTURE THE FLAG

The History of Computer Science

7th April 12:00 - 9th April 18:00

Free Entry

Open to all universities

Sign up here!

How to Play a CTF

What is a CTF?

- A Capture the Flag hacking competition
- You will solve various challenges and look for a flag, usually a string related to the challenge such as SESH{F4k3_fl4g_her3}
- Sometimes the flag won't have the SESH{ } wrapper, and you'll have to add it yourself
- Challenges require some creative thinking, but many of them should be doable if you've attended SESH sessions!

What will we cover today?

- Introduce you to some common challenge categories
- Tips, tricks, and tools
- Where to look for flags in certain challenge types
- Not all of this will be relevant for our CTF - but some of it might be…

General Challenge Categories

Web: Anything web application-hacking related!

Crypto: Cryptography-based challenges - often finding the flaw in a badly implemented algorithm (RSA is a usual favourite) but sometimes you may find some more basic ciphers to decode

Reversing: Reverse Engineering challenges, usually finding secrets in a compiled binary or looking at how to break a vulnerable function

Pwn: Usually binary exploitation challenges

- You will usually get an IP + port that hosts the vulnerable binary
- You may get access to the source code

Forensics: usually searching through log files, pcap files, and machine memory images to find secrets or evidence of malware

General Challenge Categories

Boot2Root: go from an IP address to full root/SYSTEM access!

- Usually multiple stages: foothold (likely via a web app or file sharing services), initial shell, then privilege escalation - similar to things like HackTheBox

Other:

- Steganography
- Cloud-based challenges
- Hacking Kubernetes/Docker
- Challenges hidden *on* the platform itself!
- OSINT challenges where you have to find public information on the internet

- Automation/scripting-based challenges e.g. scraping a website

What Skills Might you Need?

GOOGLE!! Google, google, google - if it exists, someone will have a CTF writeup about it

Web hacking skills:

- LFI, SQLI, XSS are common challenges!
- You may need to analyse Javascript code and look for secrets, functions that are vulnerable, obfuscated passwords, etc
- Common paths to flags and things to remember: check page source, robots.txt
- Always worth running SQLMap - but usually brute force is not the answer
- Some Challenges require Remote Code Execution - once you have the flag, you shouldn't attack the underlying infrastructure for the CTF!
- Remember your basics! Check version numbers, try admin:admin and default creds...

Crypto:

- Know your basic substitution ciphers (Caesar, Vigenere) and how to attack them with frequency analysis
- Know some common pitfalls in RSA! E.g. small primes

What Skills Might you Need?

Other quick web hacking checks:

- Nikto and wpscan
- Check for obvious paths like /admin, /wp-admin

Pwn: Remember your Buffer Overflow training from last week!

Steganography:

- Information hidden in files, such as images
- Useful commands: strings, hexdump, binwalk
- https://fareedfauzi.gitbook.io/ctf-checklist-for-beginner/steganography

Look at writeups: https://ctftime.org/ and useful tools: https://github.com/apsdehal/awesome-ctf

What Skills Might you Need?

Forensics & Reversing (often some overlap):

- Know how to analyse a pcap file with Wireshark
  - wireshark [file.pcap]
  - Look for interesting requests (e.g. to /login) or leaked secrets in handshakes
  - Right Click > Follow TCP Stream to see the full request rather than the individual packets
  - Practice: https://tryhackme.com/room/wireshark + https://tryhackme.com/room/c2carnage
- Interact with a machine image using volatility
  - volatility imageinfo -f [file.mem] to get machine details and pick a profile e.g. Win7SP1x64
  - volatility -f [file.mem] — profile=[PROFILE] pslist to list processes
- Analysing word documents: oletools -> olevba [FILE]
- Have a read of the following to see the range of possible challenges in Forensics + Reversing! https://darkdefender.medium.com/write-up-memory-forensics-in-the-def-con-dfir-ctf-c2b50ed62c6b and https://yan1x0s.medium.com/htb-x-uni-ctf-writeup-forensics-d3d122a71e36

What Skills Might you Need?

Automation:

- Using Web Scraping to visit websites and fill out forms or scrape sites for information in a short amount of time.
    - Web Scraping tools include:
        - Selenium [Link to previous session slides]
        - BeautifulSoup [Link to previous session slides]
    - When Web Scraping also remember to use XPaths in order to easily select items on a webpage.
    - If a site does have a captcha this can be counteracted by having the program wait until you press enter etc, and then letting it launch the scraping program. [Maybe also include a delay so you don’t get blocked for too many requests]
    - When web scraping an import thing to remember is that website cookies will in most cases need to be preserved.

What Skills Might you Need?

Remember your common ports:

- 20 + 21 (FTP)
- 22 (SSH)
- 80 (HTTP)
- 443 (HTTPS)
- 139 + 445 (SMB)
- 8080 (Often a web proxy)
- 3000 (Often for Node/Express servers)
- 3306 (MySQL)

This is useful if you're port scanning, but that's only usually required on a Boot2Root Challenge - most challenges will give you an IP and specific port to attack, please don't scan the rest of the ports on the machine unless you are certain it's required for the challenge!

Reversing Checklist

Commands to run:

- file [BINARY]
- Strings [BINARY]
- oletools
- ptrace

Run the binary to see what it does!

Live debug with gdb/ollydbg - Dynamic Analysis

Reverse with Ghidra - Static Analysis

- Get Ghidra here
- Try to label/retype functions and variables that you find
- You can cross-reference memory addresses with GDB to find the value of variables when the binary runs

Using Ghidra

Run with /path/to/ghidraRun

New Project

Press ‘i’ to import a file

Double click the imported file

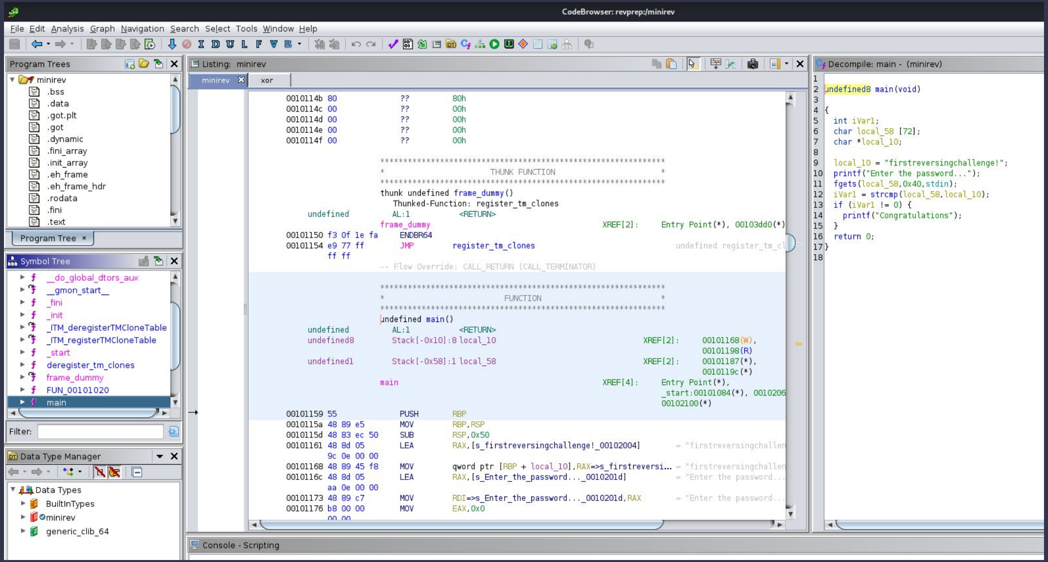
Click Analyse

Middle is assembly code, right is decompiled

Functions view lets you find main()

Search > Strings finds strings

Double click a variable/function to view it in assembly - L to rename, Ctrl + L to retype



```
0010114b 80            ??         80h
0010114c 00            ??         00h
0010114d 00            ??         00h
0010114e 00            ??         00h
0010114f 00            ??         00h

                     **************************************************************
                     *                      THUNK FUNCTION                        *
                     **************************************************************
                     thunk undefined frame_dummy()
                       Thunked-Function: register_tm_clones
       undefined       AL:1           <RETURN>
                     frame_dummy                           XREF[2]:     Entry Point(*), 00103dd0(*)
00101150 f3 0f 1e fa    ENDBR64
00101154 e9 77 ff       JMP        register_tm_clones                   undefined register_tm_cl
         ff ff
                     -- Flow Override: CALL_RETURN (CALL_TERMINATOR)

                     **************************************************************
                     *                          FUNCTION                          *
                     **************************************************************
                     undefined main()
       undefined       AL:1           <RETURN>
       undefined8      Stack[-0x10]:8 local_10                 XREF[2]:     00101168(W),
                                                                            00101198(R)
       undefined1      Stack[-0x58]:1 local_58                 XREF[2]:     00101187(*),
                                                                            0010119c(*)
                     main                                  XREF[4]:     Entry Point(*),
                                                                        _start:00101084(*), 0010206
                                                                        00102100(*)
00101159 55             PUSH       RBP
0010115a 48 89 e5       MOV        RBP,RSP
0010115d 48 83 ec 50    MOV        RSP,0x50
00101161 48 8d 05       LEA        RAX,[s_firstreversingchallenge!_00102004]       = "firstreversingchallen
         9c 0e 00 00
00101168 48 89 45 f8    MOV        qword ptr [RBP + local_10],RAX=>s_firstreversi... = "firstreversingchallen
0010116c 48 8d 05       LEA        RAX,[s_Enter_the_password..._0010201d]          = "Enter the password...
         aa 0e 00 00
00101173 48 89 c7       MOV        RDI=>s_Enter_the_password..._0010201d,RAX       = "Enter the password...
00101176 b8 00 00       MOV        EAX,0x0
         00 00
```

```
undefined8 main(void)

{
  int iVar1;
  char local_58 [72];
  char *local_10;

  local_10 = "firstreversingchallenge!";
  printf("Enter the password...");
  fgets(local_58,0x40,stdin);
  iVar1 = strcmp(local_58,local_10);
  if (iVar1 != 0) {
    printf("Congratulations");
  }
  return 0;
}
```

Mini Reversing Challenge

Challenge 1: Find the password inside the file!

Challenge 2: Decipher the password based on the method of checking it

Download files: https://shefesh.com/assets/demos/revpractice.zip

Old CTF Solutions

If we have time, I’ll go over some old challenges from last year’s CTF

Upcoming Sessions

What’s up next?

www.shefesh.com/sessions

Next week

The week after that

And the week after that

And the week after that

And the week after that

Any Questions?

www.shefesh.com

Thanks for coming!