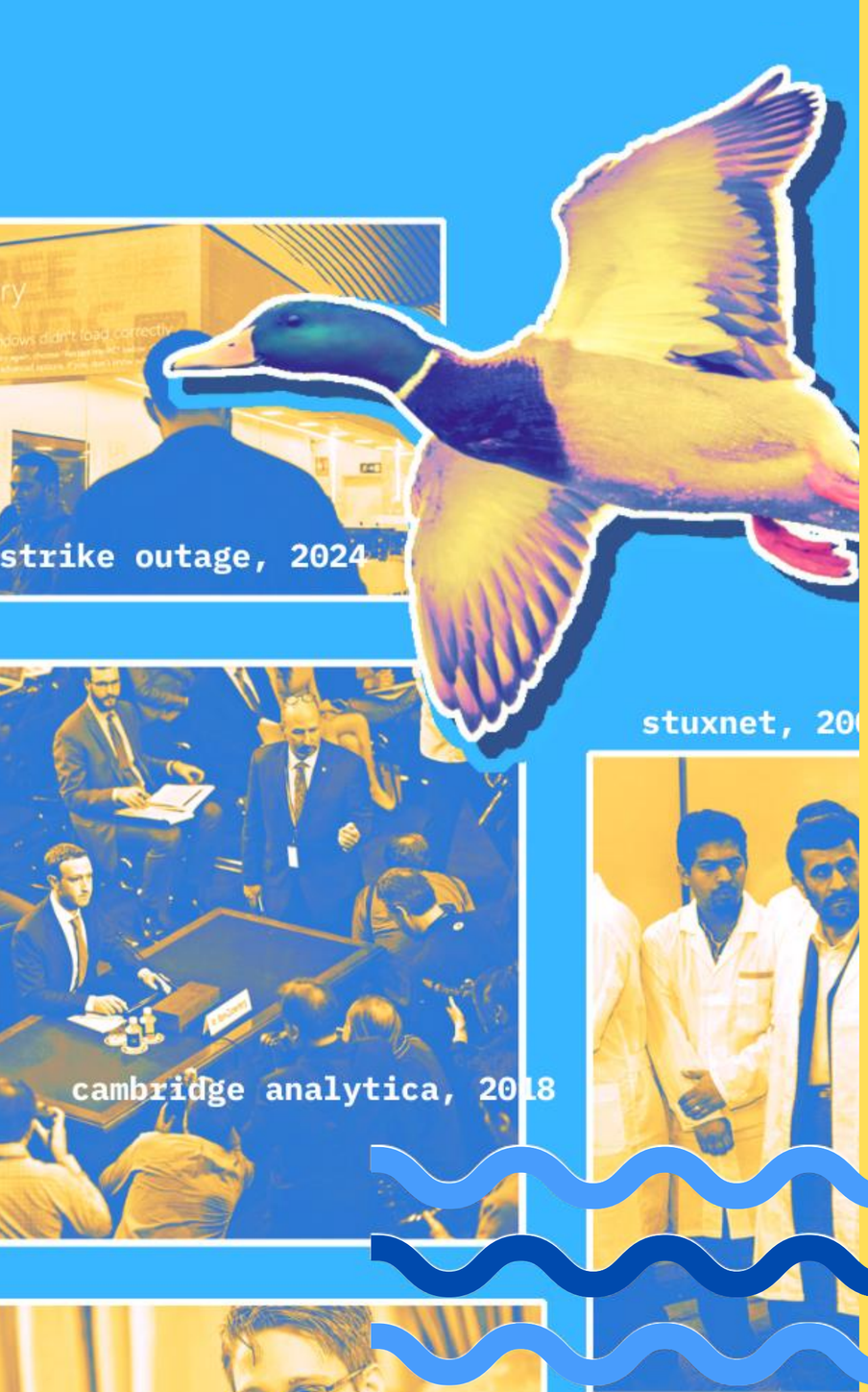


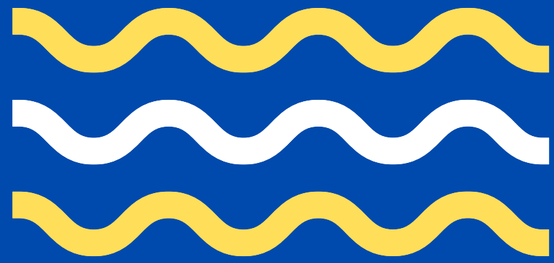
UCL CYBER SECURITY SOCIETY

CTF FUNDAMENTALS

20 November 2024

OVERVIEW

1. Intro
2. General Setup
3. Categories
4. General Approach



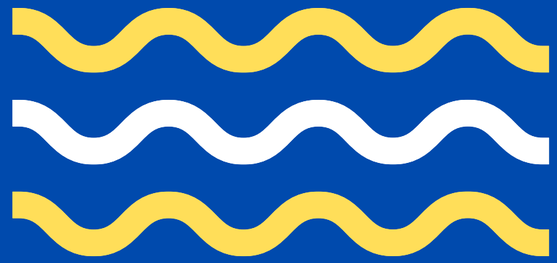
INTRO

WORKSHOP SERIES

- Goal: Learn about cyber security and develop practical skills
- Split into 4 categories:
  - Foundations
  - Vulnerabilities and Exploits
  - Defensive Security
  - Forensics
- Host small CTFs to apply what you learned
- Participate in CTFs
- Official UCL CTF team

GETTING STARTED

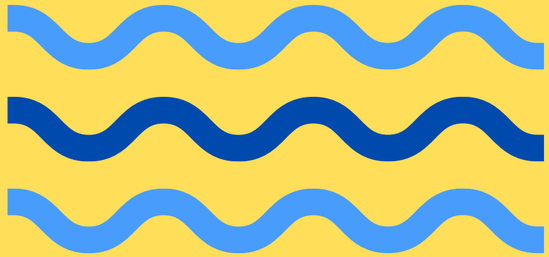
- You don’t need any cyber experience to get started!
- Start with basics and easy challenges
- Try out different categories
  - What are you most interested in?
  - What are you best at?
- Usually, people stick with 1 or 2 categories
- Read write-ups
- Create your own write-ups!



GENERAL SETUP

LINUX

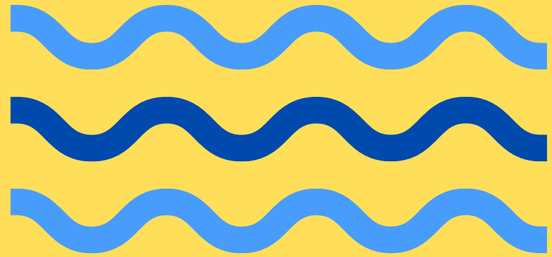
VIRTUAL MACHINE



Pros	Cons
Safe, isolated environment	Performance overhead
Revert to snapshots if something breaks	Limited access to hardware
Easily switch between distributions	Steep learning curve
Transfer VM to another machine	Disk space requirements
Custom resource management	Limited network access
Reusable setup	Extra setup to move files
Clean host system	

LINUX

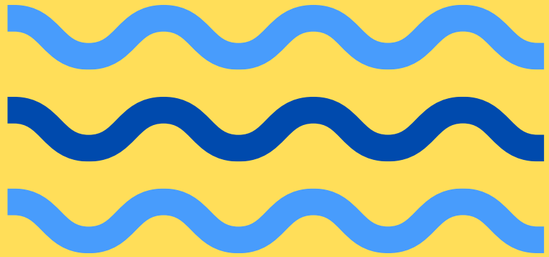
WSL



Pros	Cons
Seamless integration	No full kernel access
Easy file access	Less customisable
Lightweight	Limited hardware access
Easy setup	Doesn’t provide full isolation
Fast performance	

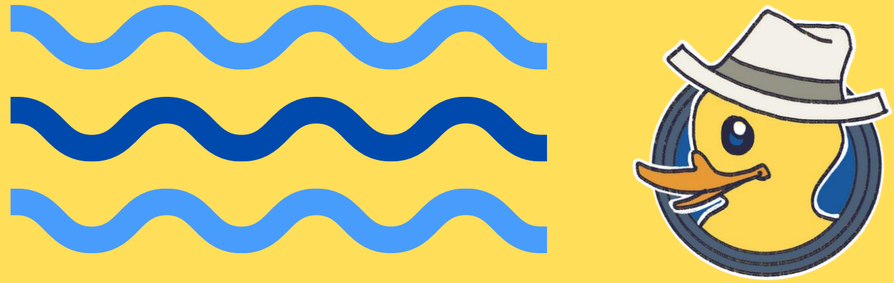
LINUX

DUAL BOOTING



Pros	Cons
Full access to hardware	Reboot required to switch
Full Linux environment	Complex setup
Customisable	Can only use one OS at a time
Isolation of OSs	Not beginner-friendly

COMMAND LINE



- echo
- base64
- hd
- cat
- file
- grep
- net cat
- strings

PYTHON ENVIRONMENTS

- Isolated spaces to install and manage packages
- Independent of system's default python setup
- Dependencies are contained and can be easily adjusted

- **pyenv**: manage multiple python versions and switch between them
- **pipx**: install and run python applications in isolated environments
- **poetry**: package manager to manage dependencies and virtual environments

DOCKER

- create, deploy, and run applications in isolated environments called containers
- containers package all the dependencies, libraries, and tools needed to run a program
- ensures consistency across different systems

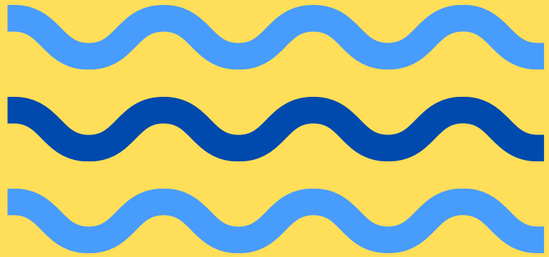
CATEGORIES

REV

- Goal: analyse and understand binaries without access to source code
- Skills:
  - Analyse assembly
  - Understand binary file structures and formats
  - Familiarity with architecture-specific instructions
  - Control flow analysis
- Common tools:
  - Ghidra or IDA Pro
  - GDB

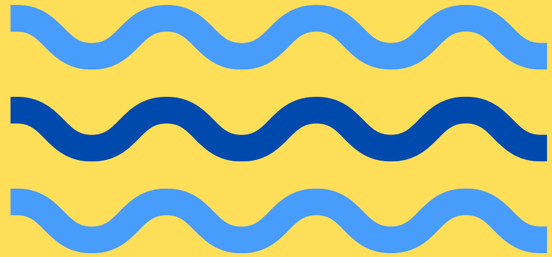
Write you own programs and look at them in Ghidra!

PWN



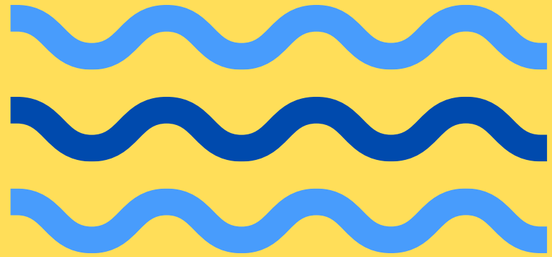
- Goal: identify and exploit vulnerabilities in binaries
- Skills:
  - Knowledge of memory management
  - Stack and heap behaviour
  - Analyse assembly
  - Scripting
  - Familiarity with common vulnerabilities and OS defenses
  - Understanding syscalls
  - Low-level program behaviour
- Tools:
  - pwntools
  - GDB

WEB



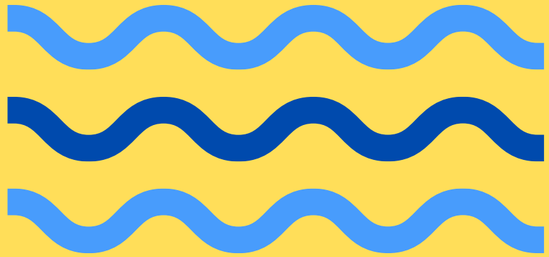
- Goal: find and exploit vulnerabilities in web applications
- Skills:
  - Understanding HTTP protocols
  - HTML
  - JavaScript
  - Databases
  - Knowledge of common vulnerabilities and misconfigurations
  - Familiarity with common web frameworks
- Tools:
  - Burp Suite
  - Postman

CRYPTO



- Goal: analyse cryptographic algorithms and break weak encryption
- Skills:
  - Understanding of cryptographic primitives and algorithms
  - Hash functions
  - Cipher modes
  - Key management
- Tools:
  - Hashcat
  - John the Ripper
  - pycryptodome

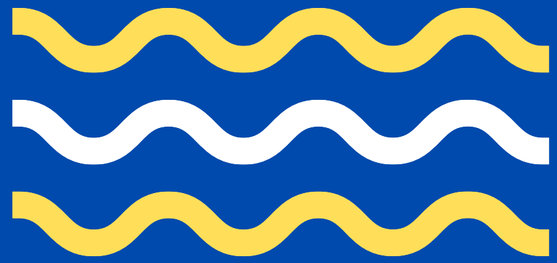
FORENSICS



- Goal: extract hidden or embedded information from files, network traffic or memory dumps
- Skills:
  - Knowledge of file formats
  - Metadata analysis
  - Steganography
  - Network traffic analysis
- Tools:
  - Binwalk
  - Steghide
  - Wireshark

MISC

- Good luck :)



GENERAL APPROACH

GENERAL APPROACH

- Read challenge description and look for any hints
- Download and inspect provided files
- Look for obvious clues (e.g. metadata, file format, strings)
- Pick the right tool
- Test any potential exploits and look at results
- Collaborate with teammates
- Keep notes!
- Write up your solution!
- Time management: know your strengths and start with challenges that seem most solvable

THANK YOU FOR COMING!!

Feedback Form