MISC

BACK TO BASICS

What are we doing today?

- Short discussion on how to approach learning security
- Overview of basic tools/techniques/ideas you should try to get familiar with
- Have a go at some MISCCTF challenges
- Eat pizza

How do I get better?

- Understand why you want to get better
- Gather resources that are relevant to your interests
- Work on improving your research skills
- Play a lot of CTF

Random assortment of basic concepts: 1 – encodings

Common examples (and how they look):

- base64
  - Take bits and group by 6
  - Encode each 6 bit grouping as a printable character
  - Pad with `=`
  - Looks like: `TUlTQyBmb3IgdGhlIHdpbg==`
  - Now we can transmit arbitrary data with printable characters

Random assortment of basic concepts: 1 – encodings

- hex
    - represent each byte as a hexadecimal number from `0x00` to `0xff`
- common theme: take bitstream and divide into chunks, then use some numerical base to represent each unique chunk.
- So be aware that many different encodings are possible.

Random assortment of basic concepts: 2 - Files

- We have an intuitive sense of what a file is, but what makes a file?
- In unix, everything is a file! (eg directories, devices)
- If a file contains data, how can we tell what kind of data?
- Use the `file` command. (Works for any file, for instance try `file .` or `file /dev/sda1`)
- Reads through a data stream and looks for 'magic bytes'

Random assortment of basic concepts: 2 – magic bytes

- many files in unix indicate their type by containing a special signature in their initial bytes.
- For instance for a file to be a JPEG, it must contain `FFD8` in its initial position
- A utility like binwalk can be used to walk through an entire data file looking for such signatures
- Useful if someone has hidden a file by simply sticking it onto the end of another, or if a file is contained within another for whatever reason

Random assortment of basic concepts: 2 - Strings

- Often we need to look for strings in files.
  - Find a specific string/pattern in a text file
  - Find printable text in a non text file
- To find a pattern in a text file, use `grep`
- To find strings in a non text file, use `strings` (then search through it with `grep`!)
- Eg `$ strings unknownfile | grep MISCCTF` to search for all strings in an unknown file that contain the pattern MISCCTF
- Can use whatever regular expression we want.

Random assortment of basic concepts: 2 – Web challenges

- Questions to get you started:
  - What is the underlying technology/framework/language?
  - Is there anything that shouldn't be in a production environment?
    - backups, git, unreferenced files alluded to in robots.txt file etc
  - Understand and test session management and authorisation
  - Trace the behaviour of any user input and input validation

More tips?

- In general, how do I learn about more tips like this?
- Look for beginner CTFs
- Read Writeups!
- Practise overcoming problems you don't have the solution to!
- Learn from others (the point of this club)

Why CTF?

- Gamification makes difficult tasks interesting and fun
- Provides a consistent format for learning new techniques
- Allows you to practise security techniques legally
- People create writeups for challenges they have solved (in theory)
- MISCCTF bot now has a writeup feature

How CTF?

- Take time to understand the underlying technology
- Look for irregularities. This can be tricky when dealing with something you haven't seen before, but this highlights the importance of practise
- So practise a lot! This is a lot easier when you are part of community that plays a lot of CTFs (This one)
- Read a lot of writeups, then work back through challenges after reading the solution.
- Today's workshop is very much about working on challenges on the bot. All the challenge authors are in this room, so take the change to pick their brains for tips and tricks.

Challenge Demo