Exploits and Mitigations

1 Buffer Overflows

- When writing data to a buffer, you overrun into adjacent memory locations
- Often results in a crash, but sometimes can be exploited for other malicious behaviour, such as gaining elevated privileges
- Can occur on the stack (stack smashing) or on the heap (heap overflow)

1.1 Protection against buffer overflows

- Detect and abort before malicious behaviour occurs
- Use input that won't allow you to overflow, such as `fgets`

2 Heartbleed

Buffer over-read vulnerability in OpenSSL

- Open source code that handles a large proportion of the world's secured web traffic
- Clients send heartbeats to server (are you alive?)
- Server responds with data
- A particular version of OpenSSL didn't check for over-read
- Each heartbeat could reveal 64k of application memory

3 Stack smashing

- What if we jumped to somewhere else where we had malicious code?
  - If we can use this on a program that has higher privilege than ourself, we can jump to deployed shellcode for that level of privilege
  - Shell code is executable code inserted as a payload for insertion attacks
- Countermeasures
  - Check buffer lengths
  - Use heap memory
  - Use ASLR
  - Use a canary

3.1 Canary Value

- Generate random number just before the stack return pointer
- After writing to buffer check if the value is same as randomly generated value
- If it has been tampered with, exit

4 Heap smashing and NOP slides

Heap memory rarely contains pointers that influence control flow:

- Needs to be combined as part of larger attack

Heap sprays

- Attempts to put a certain sequence of bytes at a predetermined location in the memory by allocation large blocks on the process' heap and filling the bytes in these blocks with specific values
- NOP slides/sleds
  - Move onto next instruction - put loads of x90's followed by your shell code. Then your return address is likely to hit one and slide to the malicious code

5 Timing attacks

Programs take slightly different amounts of time based on if the input is partially correct or not.

This can then be used to determine if each character in the password is correct, causing a drastic decrease in the number of combinations that need to be checked

6 Impact of AI on Cyber Security

- Gaining recent research traction (especially in 2019)
- Lots of unethical research taking place in this field
  - Predicting sensitive information about people
- Could make better anti-virus, or better viruses

Adversarial against humans

- Chatbots
- Can't distinguish human from AI voice
- Reinforcement learning
- Deep learning fools CAPTCHA