Buffer Overflows

Objectives

- Define what buffer overflows are
- Describe how shellcode is used into buffer overflows

What are buffer overflows?

- Applications run in memory
- Memory locations can contain information, variables or program data
- If data is overwritten through poor programming, other code can be injected causing other program or operating system access
- NIST Glossary of Key Information Security Terms defines a buffer overflow as:
- “A condition at an interface under which more input can be placed into a buffer or data holding area than the capacity allocated, overwriting other information. Attackers exploit such a condition to crash a system or to insert specially crafted code that allows them to gain control of the system.”

Basics

- A programming error allows data to be overwritten beyond the intended length of the buffer
- This can overwrite other memory locations
- Can lead to corruption of data
- Privilege escalation

How buffer overflows are found

- Testing or fuzzing
- Reverse engineering of code
- Looking at program execution
- Once a vulnerability is found, the attacker can put their own data in

Shellcode

- Shellcode is code that is used by an attacker to usually gain access to part of the operating system
- It’s used in the buffer that is overwritten
- An attacker must understand how to use the shellcode and what the underlying architecture is in order to exploit

Targeted services

- System services
- Network services
- Common libraries

What can a shell do

- Launch remote sessions
- Launch reverse remote sessions – more common if you can’t always access the system
- Tear down other defenses such as antivirus and firewalls

Why buffer overflows so prevalent

- Coders can be lazy
- Coders may use lower level languages
- Programmers may not
  - Audit
  - Test
  - Look at performance vs security

Common defenses

- Compile time
  - Stack Protection - Stackguard
  - Safe Library use
- Run time
  - Memory randomization
  - Operating system memory protection – EMET
- How to protect your organization
  - Don’t run all the software you can!
  - Use protection mechanisms especially on servers