


DLL Injection

by Chris Grier
grier@uiuc.edu

Outline

- What DLL Injection can do
- Function Basics
- Injection
- Hijacking Functions
- Cleaning things up
- Possible Problems

DLL Injection Does...

- inserts a DLL you write into a processes memory space that doesn't necessarily want it
- provided a loader and a dll
- allows you to change and extend the way a function works *without knowing the function source*

Function Basics

- Things that you need to know:
  - Some basic assembly (jumping, calling, returning and addresses)
  - some windows api functions
  - how programs work in your computer

Injection Overview

- Get the program going that you want to inject into
- Get the DLL you want injected in somewhere in the memory space of the target program
- Initialize the DLL
- Detach and run away

Injection Details

- Start or open a process
- Allocate some space and write the DLL name to it
- use LoadLibrary() to get your dll in
  - use CreateRemoteThread() to get LoadLibrary called
  - need to use GetProcAddress
- http://www.devx.com/Intel/Article/21023

Hijacking Functions

- Be *really* careful with declarations and calling conventions
- When the program calls a specific function, you want to redirect it to your code
- How?

Hijacking Details

- Find the address of the function you want (loleasy)
- Use VirtualProtect() to change the permissions on the function area
- Write a *jmp* #evilfunctionaddress# to the beginning of the function and a *ret* after it
- Use VirtualProtect() to change it back

Hijacking: Extra Credit

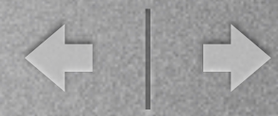
- Calling the original function after your code executes
  - Instead of inserting a jmp+ret, copy part of the orginal function out somewhere
  - insert the jump to your code
  - execute your code + the copied code
  - jmp back and finish executing the original

Hijacking: More Extra Credit

- Use the function parameters that the original function is called with
  - Requires dealing with the stack pointer and making your own local variables
- You can modify the values the original function is called with!

Cleaning things up

- Tricks
  - Hiding code inside empty spots, things are allocated in fixed sizes in memory, and there are “holes” in DLLs.
  - If you are good at disassembling, you can use parts of the original function in your function

Possible Problems

- Don't mess up the stack if you are going to call the original function
- instructions don't all have the same length
- Games have anti-cheat mechanisms
  - Lots of cool stuff here, you will learn a lot by trying to cheat in a game with prevention mechanisms

More things

- encrypted and packed functions that are decoded on the fly
- Programs to use: Dependecy Walker, Tsearch, IDA, OllyDebug, Windbg
- Ask grier@uiuc.edu for suggestions or good places to find stuff on these things.