Backdooring Win32 PE Files

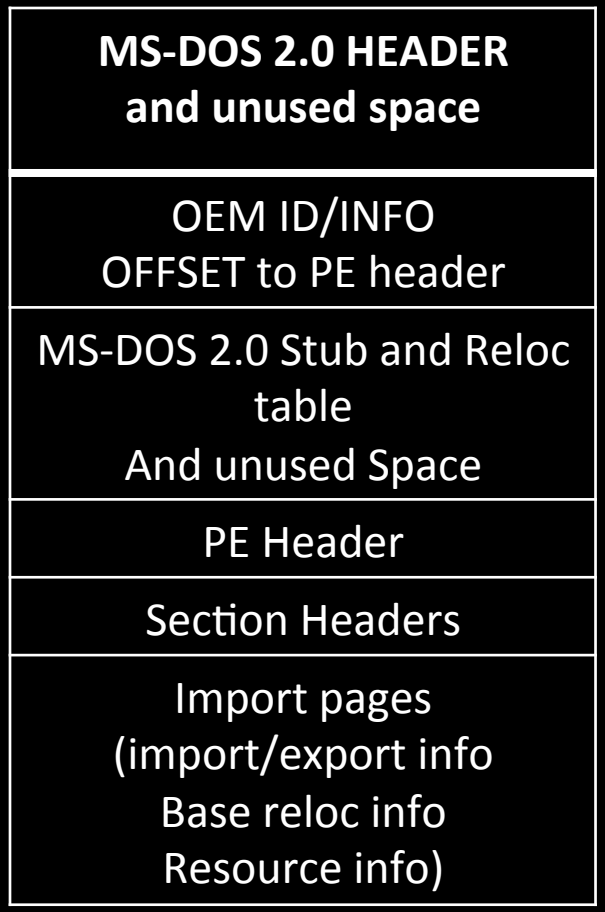
Joshua Pitts

Nova Hackers

About Me

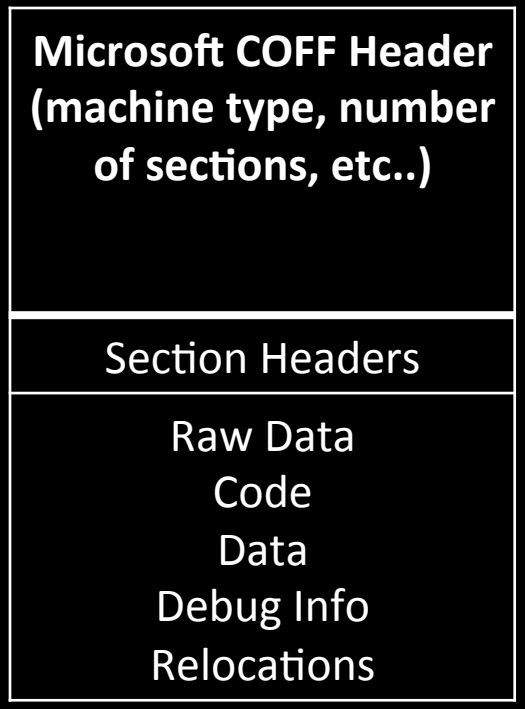
- Joshua Pitts
- Twitter://@midnite_runr
- Former US Marine (SIGINT)
- Pentesting and operational security
- I like long walks on the beach
- Certs (in order of difficulty): OSCE, OSCP, OSWP, GCFA, CISA, CISSP
- Python, Bash, Batch, C, x86 Intel ASM

The Portable Executable Format



- Not much has changed in the last 20 or so years
- Must be backwards compatible (win8 must read the header)
- Kind of weird (feels like spaghetti code)
- Easy to automatically manipulate
- http://msdn.microsoft.com/library/windows/hardware/gg463125

The Common Object File Format (COFF) Format



- This is included in the PE File Format
- The most important section for RE
- Includes:
  - Machine Type
  - Number of Sections
  - Size and Access (RWE) of Sections
- Typically includes the rest of the file Code, Data, Debug, Reloc (the actual sections)

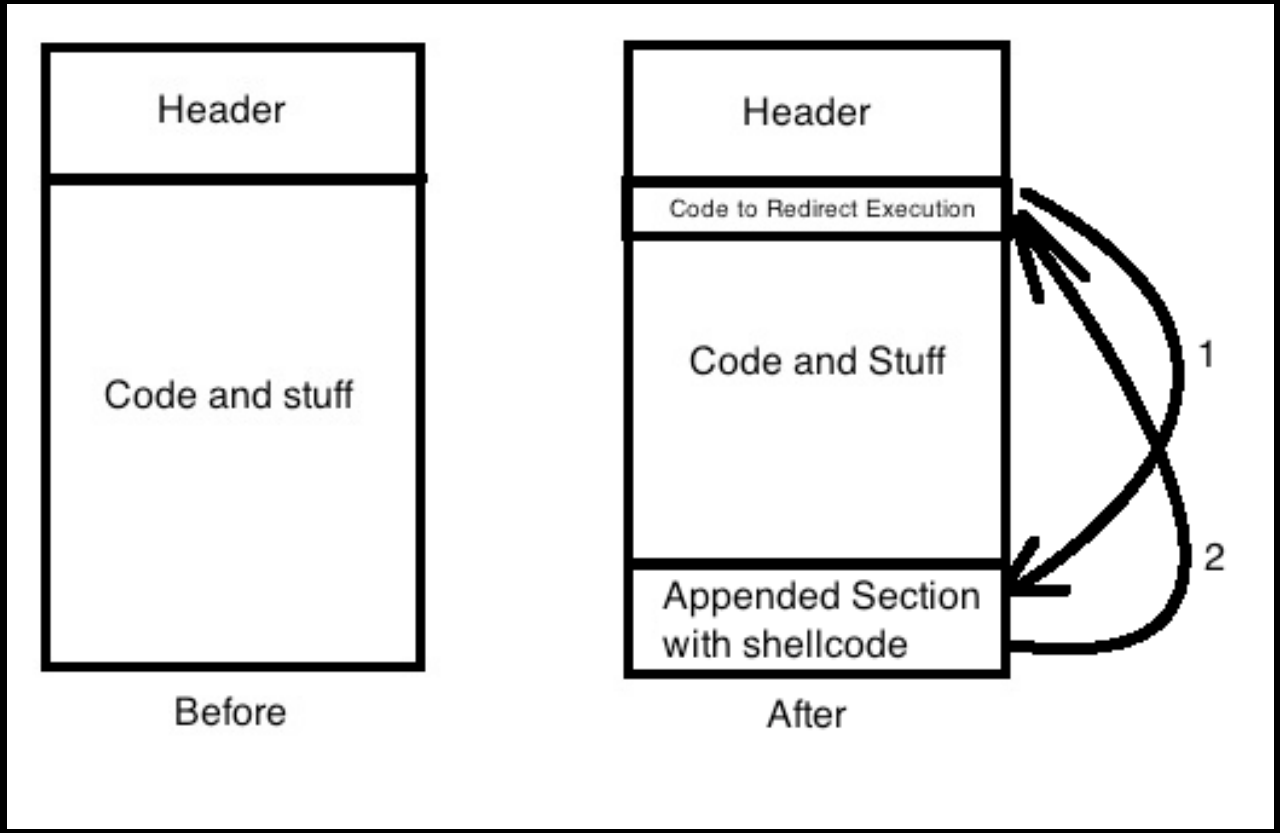
The PE File Format Can be Patched Easily

- Microsoft does it (patching?)
- Software crackers do it
- Key Gens do it
- Metasploit does it (./msfencode –t)
- Pentesters should too.. (need knowledge of ASM and basic debugging)

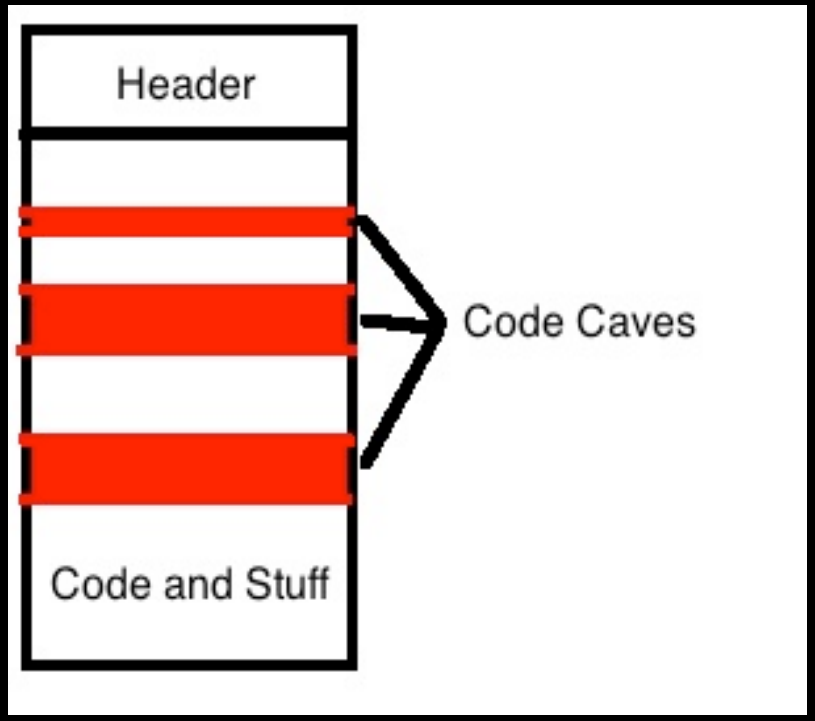
How I Learned to Patch an EXE

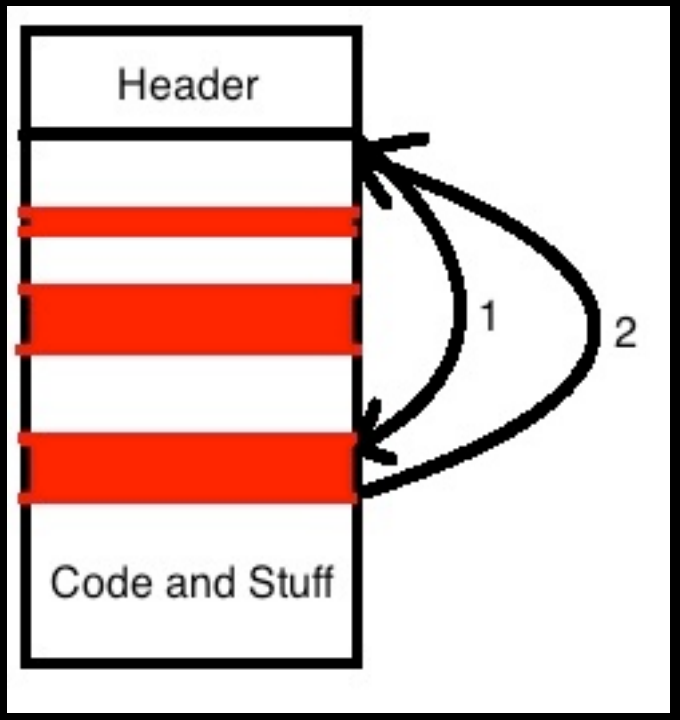
Taught by Offensive Security – Cracking The Parameter - Manually

Similar to this: http://blip.tv/nu1lsecur1ty/av-shmoocon-presentation-2008-6362110



Using Code Caves Also Works





Why should Security Pros want to patch (backdoor) exes/dlls?

- Social engineering/Penetration Testing
- Red Teaming (Persistence)
- Prototyping shellcode for A/V bypass testing
- Proactive protection
- Because it is fun
- There might be profit, though not confirmed

The Goal

We need backdoors that:

- Persist
- Hide in plain sight
- Function normally
- User will want to use them (over and over)
- Avoid Anti-Virus
- Run with System/Admin level privileges

We need to do this:

- In an automated way
- That must be customizable
- And support many formats (x32/x64)

Solution

The Backdoor Factory

(Not what you are thinking)

- A way to insert a backdoor into most win32 EXE and DLL (x64 support in the future)
- EXE/DLL attempts to continue execution after shellcode delivery
- Written in python
- Uses standard python libraries
- Different than the metasploit msfencode –t method

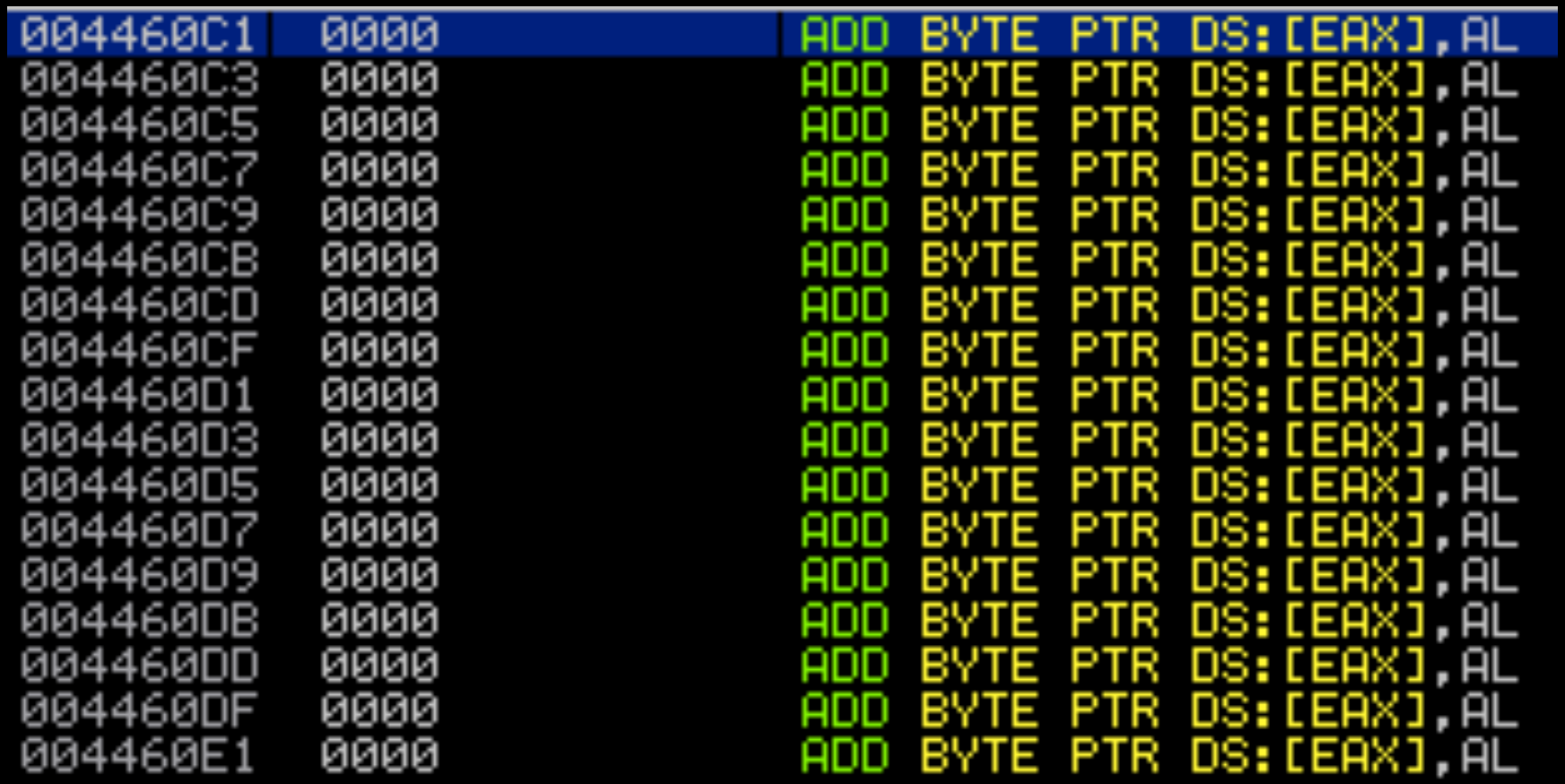
How does it work?

- Maps the PE Header (without PEFile <-TL;DR)
- Determines if it supports the EXE/DLL
- Gathers information to allow for after shellcode execution (if shellcode supports it)
- Append a code cave or find code cave to patch
- Patches binaries directly
- Encodes shellcode if the encoder scheme is built into the tool

PE File Entry at Run Time (Before)

PE File Entry at Run Time (After)

Beginning of Code Cave (Before)



Beginning of Code Cave (After)

```
004460C1   90              NOP
004460C2   90              NOP
004460C3   60              PUSHAD
004460C4   9C              PUSHFD
004460C5   FC              CLD
004460C6   E8 89000000     CALL bd_Tcpvi.00446154
004460CB   60              PUSHAD
004460CC   89E5            MOV EBP,ESP
004460CE   31D2            XOR EDX,EDX
004460D0   64:8B52 30      MOV EDX,DWORD PTR FS:[EDX+30]
004460D4   8B52 0C         MOV EDX,DWORD PTR DS:[EDX+C]
004460D7   8B52 14         MOV EDX,DWORD PTR DS:[EDX+14]
004460DA   8B72 28         MOV ESI,DWORD PTR DS:[EDX+28]
004460DD   0FB74A 26       MOVZX ECX,WORD PTR DS:[EDX+26]
004460E1   31FF            XOR EDI,EDI
004460E3   31C0            XOR EAX,EAX
004460E5   AC              LODS BYTE PTR DS:[ESI]
004460E6   3C 61           CMP AL,61
004460E8   7C 02           JL SHORT bd_Tcpvi.004460EC
004460EA   2C 20           SUB AL,20
```

Features

- Cave locator: search for caves > x bytes
- Shellcodes from metasploit: windows/ shell_reverse/bind_tcp
- Append a code cave to store shellcode
- Backdoor an entire directory of exes/dlls
- Host and port selection
- Simple XOR encoder
- Hunt and Inject (must be run on windows)
- Randomization of nops (5 types) and of ones compliment

DEMO

- Code Cave Finder
- Backdoor via appending a code cave
- Backdoor via finding a code cave
- Backdoor a directory of EXEs
- Injector
- AV Avoidance

Mitigations

- UPX, very difficult to backdoor
- File Integrity Checks
- Application whitelisting
- Run only Trusted executables
- Check your hashes
- Wipe the drive

Future

- x64 Support
- ROP gadget encoding
- Multiple cave jumping
- ELF/Mach-O formats?
- Import table patching (Evil DLLs?)
- Predetermined code cave selection for injector

Questions

- Twitter://@midnite_runr
- Code: http://bitbucket.org/secretsquirrel/