

Computer Security

Malware: Malicious Code

April 13, 2004

©2004, Bryan J. Higgs

1

Sources

- Here are some good books on the topic of *malware* and related security topics:
 - *Counter Hack: A Step-by-Step Guide to Computer Attacks and Effective Defenses*, by Ed Skoudis, Prentice-Hall
 - *Malware: Fighting Malicious Code*, by Ed Skoudis, Prentice-Hall
 - *Hacking Exposed: Network Security Secrets & Solutions*, by Stuart McClure, Joel Scambray and George Kurtz, McGraw-Hill/Osborne
 - *Web Hacking: Attacks and Defense*, by Stuart McClure, Saumil Shah and Shreeraj Shah, Addison-Wesley
 - *Web Security, Privacy & Commerce*, by Simson Garfinkel, O'Reilly and Associates
 - *Computer Security: Art and Science*, by Matt Bishop, Addison-Wesley

2

What is "Malware"?

- Skoudis, in *Malware (op. cit.)*, offers the following definition:

Malware is a set of instructions that run on your computer and make your system do something that an attacker wants it to do.

- Bishop (*op. cit.*) defines the following:

Malicious logic is a set of instructions that cause a site's security policy to be violated.

However, it's worth noting that bad things can be done even when not actually violating security policy – depending on your definition of what a (your) security policy might be!

3

What Kinds of Malware Are There?

- Skoudis lists the following types of malware:

- Virus
- Worm
- Malicious mobile code
- Backdoor
- Trojan horse
- RootKits
- Combination malware

- See:

- <http://infosecuritymag.techtarget.com/2002/jul/faster.shtml>

4

Why Did Malware Arise?

- Some people enjoy the challenge and the twisted logic involved; get a kick out of showing how "smart" they are.
- Our computing platforms are becoming very uniform
 - Just as with biological viruses, a diverse environment tends to be more resistant to contagion. But we're going in the opposite direction (WinTel).
- Software vendors have placed more focus/importance on producing software quickly than they have on making that software secure and bug-free. Not much is likely to change here.
- Software vendors have to keep adding functionality; more functionality implies more things to go wrong, less secure. We have to live with this.
- Software vendors have felt the need to add executable content to everything. Pandora's box was opened some time ago...
 - Microsoft Office macros, JavaScript, VBScript in browsers, etc.

5

Malicious Code History

- Here is a malicious code historical timeline (Skoudis, *op. cit.*):

1981-2 – first reported computer viruses discovered in games for the Apple II computer	1998 – Back Orifice, by "Cult of the Dead Cow"
1983 – Fred Cohen formally defines the term <i>Computer Virus</i>	1999 – Melissa Virus/Worm
1986 – First IBM PC virus, Brain, infected MS-DOS systems	1999 – Back Orifice 2000
1988 – The infamous Morris Internet Worm released (accidentally?)	1999 – Distributed Denial of Service Agents
1990 – First Polymorphic Virus (appearance-altering virus)	1999 – Knark Kernel-Level Toolkit released
1991 – Virus Construction Set (VCS) released	2000 – Love Bug shut down tens of thousands of systems around the world
1994 – Good Times Virus Hoax	2001 – Code Red Worm. Over 250,000 machines fell victim
1995 – First Macro Viruses	2001 – Kernel Intrusion System (for Linux)
1996 – Netcat released for UNIX	2001 – Nimda Worm
1998 – First Java Virus (StrangeBrew)	2002 – Setiri Backdoor
1998 – Netcat released for Windows	2003 – SQL Slammer Worm. Disabled several ISPs in South Korea, and problems worldwide.
	2003 – Hydan Executable Steganography Tool. Allows data to hide inside executables.

6

Viruses

- Skoudis defines a virus thusly:
A virus is a self-replicating piece of code that attaches itself to other programs and usually requires human intervention to propagate.
- Bishop defines it as:
A computer virus is a program that inserts itself into one or more files and then performs some (possibly null) action.

7

Virus Targets

- A virus can infect:
 - An executable file
 - On Windows, these are .COM and .EXE files, but other types of files can be infected, also.
 - A disk's boot sector
 - Document files, taking advantage of *executable content*:
 - Microsoft Office
 - Word
 - Excel
 - PowerPoint
 - WordPerfect Office
 - StarOffice
 - AutoCAD
 - Scripts (source modifications)
 - Windows Scripting Host, VBScript
 - UNIX shell scripts, Perl
 - PHP

8

Multipartite Viruses

- A virus that is not limited to a single target is called a ***multipartite virus***.
- Often, this means that a virus can infect either boot sectors, or executables, or both.
- There is nothing to stop a single virus from infecting any combination of the previously described targets.

9

Virus Propagation Mechanisms

- Here's how a virus can propagate from one system to another:
 - Removable storage
 - Floppies, ZIP drives, etc.
 - CDs, CD/Rs, CD/RWs, and DVDs are more difficult, although some vendors have (accidentally!) distributed infected product CDs
 - eMail and Downloads
 - Usually, this requires a user to explicitly open an attachment, or similar.
 - However, content displayed in an HTML-capable email client, such as Outlook, can be exploited.
 - Shared directories
 - Windows file sharing using Server Message Block (SMB) protocol
 - Network File System (NFS) shares
 - Peer-to-peer services like Gnutella, Kazaa, etc.

10

Defending Against Viruses

- No single tool will protect against all malware attacks. You need to employ multiple levels of protection:
 - AntiVirus Software (**you are using one of these, right?**)
 - On user workstations, file servers, mail servers, application servers, even handhelds.
 - Norton AntiVirus, McAfee AntiVirus, AVG AntiVirus (<http://www.grisoft.com/>)
 - **Remember to keep your Antivirus database up to date!**
 - Configuration Hardening
 - Principle Of Least Privilege (don't give any more access than you need)
 - Minimize number of active components (disable functionality you don't need)
 - Turn off Microsoft Office macros unless (and only when) absolutely needed
 - User Education
 - Sometimes your weakest link
 - Educate your users to know what to look for and what to avoid.

11

Worms

- Skoudis defines a worm as follows:
A worm is a self-replicating piece of code that spreads via networks and usually doesn't require human interaction to propagate.
- Bishop defines it as follows:
A computer worm is a program that copies itself from one computer to another.

The essential difference between a worm and a virus is that a worm can propagate without human interaction.

The attributes of both a worm and a virus can be combined in a single piece of malware, and have been – for example, the Melissa virus/worm.

12

The Advantages of Worms

- By adding the property of self-propagation, a worm makes itself much more powerful
- It certainly will spread much faster – sometimes at amazing rates.
- Advantages:
 - Tracing the originator becomes much more difficult.
 - Many thousands of machines can be compromised in a very little time.
 - Amplifies damage
 - Can provide a divide and conquer solution to a problem, such as trying to break user passwords, etc.

13

History of Worms

- Here are some major worm events:
 - 1521 – Diet of Worms – Martin Luther; Protestant Revolution
 - 1988 – Morris Worm ("The Internet Worm")
 - Written by Robert Tappan Morris. He claimed it was a benign experiment that got out of control. Was tried and convicted, and sentenced to 3 years' probation. He is now a professor at MIT.
 - His father, Robert Morris, Sr., was one of the originators – around 1962 -- of a computer game called Darwin, which was the first to incorporate many of the ideas behind viruses and worms. At the time his son wrote the worm, the father was the chief security expert of the National Security Agency (NSA). Whoops!
 - 1999 – Melissa
 - Microsoft Word macro virus
 - Spread via Microsoft Outlook email
 - Both a virus (infecting .doc files) and a worm (spreading via network)
 - 2000 – The Love Bug
 - Spread via Microsoft Outlook email.
 - 2001 – Ramen, which targeted Linux systems

14

History of Worms (cont.)

- 2001 – Code Red
 - Targeted Microsoft Windows IIS Web Server
 - Planned a packet attack against www.whitehouse.gov
- 2001 – Nimda
 - Targeted Windows (Internet Explorer, file sharing, IIS Web Server, Outlook)
 - Employed 12 different propagation mechanisms
 - Released only a week after the Sept 11th attacks
- 2002 – Klez
 - Targeted Outlook and Windows file sharing
 - Attempted to disable Antivirus products
- 2002 – Slapper
 - Targeted Linux systems running Apache with OpenSSL
 - Exploited a flaw in the OpenSSL code used by Apache
- 2003 – SQL Slammer
 - Targeted Windows systems running Microsoft SQL Server
 - Disabled South Korea's Internet connectivity for several hours
 - Shut down thousands of ATMs in North America

15

Components of a Worm

- Skoudis lists the following components of a worm:
 - Warhead
 - Breaks into/gains access to victim machine
 - Buffer overflow, file-sharing, email, misconfigurations, etc.
 - Propagation Engine
 - Transfers remainder of the worm's body to its target
 - FTP, HTTP, SMB, etc.
 - Target Selection Algorithm
 - Looks for new victims to attack
 - Email addresses, host lists, trusted systems, etc.
 - Scanning Engine
 - Actively scans addresses found by the target selection algorithm
 - Payload
 - Implements some specific action on target machine
 - Opening up backdoor, planting a DDOS flood agent, cracking a cipher key/password

16

Ethical Worms

- Some people have thought about using worm technology (if we can call it that) to implement useful things such as:
 - Automatically patching systems
 - Is Microsoft Windows Update an ethical worm?
 - Symantec LiveUpdate?
 - "White worms" fight to keep the bad worms out
 - What color hat are you wearing today kemosabe? 
 - <http://www.tcnj.edu/~hofmann/kemosabe.htm>
- The idea is basically considered too dangerous to take seriously.

17

Worm Defense Strategies

- Belt and suspenders approach:
 - AntiVirus
 - Keep vendor patches up to date on all systems
 - Harden publicly accessible systems
 - Block arbitrary outbound connections
 - Establish incident response capabilities

18

Malicious Mobile Code

- Skoudis provide the following definitions:

Mobile code is a lightweight program that is downloaded from a remote system and executed locally with minimal or no user intervention.

Malicious mobile code is mobile code that makes your system do something that you do not want it to do.

19

Browser Scripts

- Scripts (JavaScript, JScript, ECMAScript, VBScript) that are placed in a web page, and are executed in the context of the browser.
- Can cause:
 - Resource exhaustion
 - <http://www.rivier.edu/faculty/bhiggs/web/cs572aweb/Tools/Exhaust.htm>
 - Browser hijacking
 - <http://www.rivier.edu/faculty/bhiggs/web/cs572aweb/Tools/Trap.htm>
 - Stealing Cookies
 - Scripts can access cookies (sometimes via buggy browser code)
 - Cross-site Scripting (XSS) Attacks

20

ActiveX Controls

- Microsoft implements a Component Object Model (COM) which allows one application to access another application's code
 - For example, you can embed Excel spreadsheet cells in a Word document, etc.
- ActiveX Controls are special COM objects that are designed to be downloaded and used within Web pages.
 - Compiled programs which are far more powerful than scripts
 - Once running, can do anything a regular program can do:
 - Access files, registry
 - Connect to the network
 - Invoke other programs
 - etc...

21

ActiveX Control Security

- Microsoft added a level of security to ActiveX controls:
 - A control can be cryptographically signed, using Microsoft's *Authenticode*[®] -- that is, using digital certificates.
 - This merely says that the signing authority for the certificate states that the certificate was issued for the specified user.
 - Whether you trust that user, and any of the ActiveX controls signed with that user's certificate is up to you.
 - A web page that tries to activate a signed ActiveX Control may (depending on the browser's security settings) prompt you for whether you will accept (i.e. trust) this control.
 - If you accept it, then it can run with any and all of your privileges
 - If you don't accept it, then it is not allowed to run at all.
 - The granularity of security is very coarse.

22

ActiveX Control Security

- An ActiveX control (whether signed or not) can be flagged as 'safe for scripting' or not.
 - If it is not flagged as safe for scripting, then it cannot be run from a browser script.
 - Even ActiveX controls which have been designated safe for scripting have been found to have bugs and security holes which can be (and have been) exploited.
- There are malicious ActiveX controls (often invisible ones) out there, that bad guys try to trick you into accepting
- In .Net, Microsoft has augmented the security model to allow for a finer granularity of security, similar to Java's security model.

23

Spyware Browser Plug-ins

- A browser plug-in is some code that extends the browser's capabilities.
 - Plug-ins written for Microsoft Internet Explorer are known as Browser Helper Objects (BHOs)
 - Many are available for free
 - For example, browser toolbars from Yahoo, Google and many others
 - BHOs are a popular way for Spyware to gain access to your system
 - For a list of spyware programs and plug-ins, see www.cexx.org/adware.htm
 - There are a number of tools which allow you to discover lurking spyware/adware on your computer, and to remove them:
 - BHODemon from www.definitivesolutions.com
 - Ad-aware from <http://www.lavasoft.de/>
 - Spybot Search & Destroy from <http://www.safer-networking.org/index.php?page=download>

24

Java Applets

- Java is a platform-neutral general-purpose object-oriented programming language
 - Despite the name similarity, it has nothing to do with JavaScript.
- Java applets are pieces of Java code that can be run in web pages, typically to provide more dynamic content.
- Java is a product of Sun Microsystems
 - It competes in the browser space with Microsoft's ActiveX controls
 - There was a long, acrimonious court battle over Java between Microsoft and Sun
 - Java is a significant threat to Microsoft's hegemony in operating systems and environments, so Microsoft tries to undercut it at every opportunity.

25

Java Applets

- In order to run Java applets, a browser must:
 - Support Java at the proper version number level
 - Don't use Microsoft Internet Explorer's "Microsoft VM"; it's completely out of date, and may have some security vulnerabilities
 - Instead, install the Java plug-in for your browser, which is freely downloadable from <http://java.sun.com/products/plugin/>
 - Netscape 7.1, Mozilla and some other browsers already come with the proper version of Java
 - Java must be enabled in the browser's security settings

26

Applet HTML Tags

- To add a Java applet to a web page, you use the `<applet>` tag
- For example:

```
<applet width="100" height="50"
        code="frameInvoker.FrameInvokerApplet.class"
        codebase="../classes/" align="middle">
  <param name="FrameClassName"
        value="substitution.MonoAlphabeticSubstitutionFrame">
Your browser does not have Java support.
Please install the Java Plug-in.
</applet>
```

27

Java Applet Security

- A Java applet runs in a '*sandbox*', under the control of the browser.
- The '*sandbox*' restricts what the Java Applet is allowed to do:
 - It cannot access any files on the browser machine's disks
 - It cannot run programs on the browser machine
 - It cannot make network connections to any machine except the one from which it was downloaded.
 - and more...

28

Signed Java Applets

- Originally, all Java applets were restricted to the sandbox activities
- However, the need for more capabilities became apparent, and so the Java security model was extended:
 - Sun introduced *signed applets*, which are cryptographically signed with a certificate.
 - If a web page attempts to activate a signed applet, the user will typically be asked, via a dialog box, whether the user wants to accept (trust) the applet
 - So far, this is similar to the signed ActiveX control security model
 - In addition, Sun went further: It has a security model whereby an applet may be granted a specific subset of capabilities – there is a `java.policy` file whose contents are used to determine the permissions granted to this particular applet.

29

Java Applet Security Model

- Here is what the browser does to determine how to run a Java applet:
 - If the applet is unsigned, then run it in the highly-restrictive sandbox
 - Otherwise, it is signed, so:
 - The browser (actually, the JRE) checks the `java.policy` file to determine whether specific privileges were granted to the applet's URL. If so, the applet is run within those restrictions.
 - If the signed applet does not have an assigned security policy, the JRE checks whether the applet's author is on the list of trusted applet authors. If so, the applet is executed with full access privileges.
 - Otherwise, the JRE prompts the user to determine whether the user wishes to accept/trust the applet's author. If so, the author is added to the trusted applet authors list, and the applet is executed with full access privileges. All other applets signed by that author will subsequently be silently run with full access privileges.

30

Email Vulnerabilities

- Remember that many email clients, such as Microsoft Outlook, or Outlook Express, etc. can display HTML.
 - They are then subject to all the same vulnerabilities as web browsers
 - They may be able to run scripts, and you may or may not be able to use security configurations to restrict this.
- Very important: Never do web browsing or email reading from a superuser account (i.e. with root privilege on UNIX or with Admin privileges in Windows) **Why not?**
 - Remember the Principle Of Least Privilege !

31

Backdoors

- Skoudis defines a backdoor as:
A backdoor is a program that allows attackers to bypass normal security controls on a system, gaining access on the attacker's own terms.

32

Backdoor Access

- Backdoors focus on giving the attacker access to the victim machine. Access can include:
 - Local escalation of privilege
 - With enhanced privileges, the attacker can do essentially anything on the victim machine
 - Remote execution of single commands
 - Allows remote attacker to cause a single command at a time to be executed on the victim machine
 - Remote command line access
 - Allows remote attacker to enter commands directly into a command prompt on the victim machine
 - Remote control of the GUI
 - Allows remote attacker to see the victim machine's GUI, and take over control of that GUI

33

Backdoor Installation

- An attacker can use all the previously discussed methods to break into the target system and subsequently install a backdoor:
 - Exploit a code vulnerability, such as a buffer overflow bug
 - Use virus, worm, or malicious mobile code mechanisms
 - Trick the user into installing the backdoor directly
 - Email attachment
 - File-sharing
 - Trojan Horse techniques

34

Automatic Backdoor Restarts

- Once a backdoor has been installed and started up, what happens if the local user logs off, or the system is rebooted? Doesn't the backdoor go away?
 - The backdoor implementor can exploit the target platform's Operating System-specific mechanisms for causing programs to be automatically [re]started

35

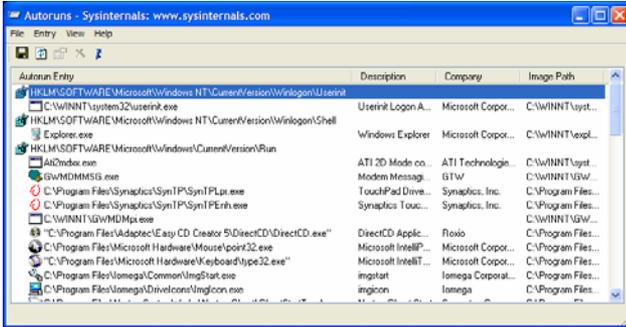
Automatic Backdoor Restarts

- Windows has lots of automatic start mechanisms:
 - Autostart directories:
 - C:\ ... \Start Menu\Programs\Startup
(exact location depends on Windows version)
 - Startup files:
 - Win.ini, System.ini, Winit.ini, Winstart.bat, Autoexec.bat, Config.sys
 - Registry entries:
 - HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows [NT]\CurrentVersion\ { RunServicesOnce, RunServices, RunOnce, Run, RunOnceEx, Winlogon\Userinit, ShellServiceObjectDelayLoad, Policies\Explorer\Run }
 - HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\System\Scripts
 - HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows [NT]\CurrentVersion\ { RunServicesOnce, RunServices, RunOnce, Run, RunOnceEx, Policies\Explorer\Run, Windows\Run, Windows\Load
 - HKEY_CURRENT_USER\SOFTWARE\Policies\Microsoft\Windows\System\Scripts
 - HKEY_CLASSES_ROOT\Exefiles\Shell\Open\Command
 - Using the Task Scheduler service

36

Automatic Backdoor Restarts

- A useful, free, tool called *AutoRuns* may be found at:
<http://www.sysinternals.com/ntw2k/source/misc.shtml#autoruns>
- It shows a list of programs which will be automatically run:



37

Automatic Backdoor Restarts

- UNIX also has its share of mechanisms for autostarting:
 - Modify the `init` daemon's `/etc/inittab` script
 - `init` starts up a number of 'services daemons', for example:
 - `httpd` -- a web server
 - `sendmail` -- a mail server
 - `sshd` -- the Secure Shell daemon
 - `ppp` -- the Point-to-Point Protocol (used for modem dial-up connections)
 - The backdoor could be added to this list, so it runs as a 'service'
 - Or one of the existing services daemons could be reconfigured to run the backdoor as part of its startup procedure.
 - Change user startup scripts
 - These include:
 - `.login`, `.cshrc`, `.kshrc`, `.bashrc`, `.bash_profile`, `/etc/profile`, `.profile`, `.logout`, `.xinitrc`, `.xsession`
 - Schedule jobs with `cron`

38

Backdoor Detection

- One way of checking for the existence of a backdoor on your system is to run a *file integrity program*
 - One venerable and popular one is *Tripwire*
 - Commercially available for UNIX and Windows at:
 - <http://www.tripwire.com/>
 - Free, open source version available, only for UNIX, at:
 - <http://www.tripwire.org/> and
 - <http://sourceforge.net/projects/tripwire/>
 - Other tools include:
 - AIDE -- open source tool from:
 - <http://www.cs.tut.fi/~rammer/aide.html>
 - <http://sourceforge.net/projects/aide>
 - Osiris -- open source tool from:
 - <http://osiris.shmoo.com/>

39

Netcat

- *Netcat* is a program that reads and writes data across network connections, using the TCP or UDP protocols
 - It can act as a listener and/or a client
 - It can be used as a backdoor for either legitimate or nefarious purposes.
 - It can be obtained for free from:
 - http://www.atstake.com/research/tools/network_utilities/
 - or a GNU version from:
 - <http://netcat.sourceforge.net/>
- There is an enhanced version of Netcat which adds encrypted communication, called *Cryptcat*, also available at:
 - http://www.atstake.com/research/tools/network_utilities/
 - <http://sourceforge.net/projects/cryptcat/>

40

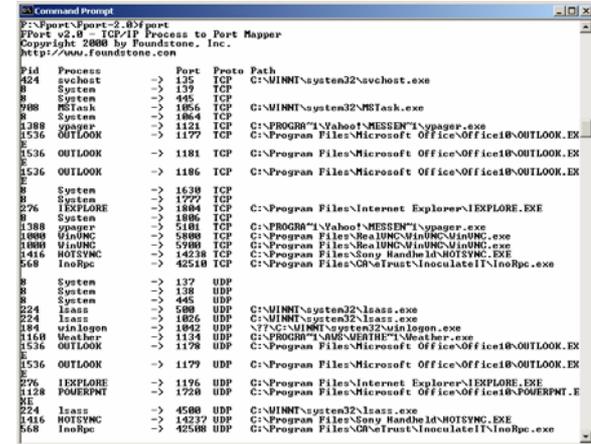
Countermeasures for Backdoors (Windows)

- **Fport**, available for free from:

<http://www.foundstone.com/index.htm?subnav=resources/navigation.htm&subcontent=/resources/freetools.htm>

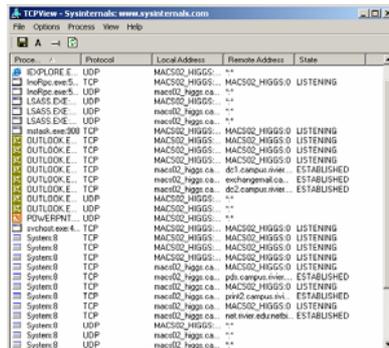
shows not only the list of ports in use, but also which programs are listening on those ports.

Countermeasures for Backdoors (Windows)



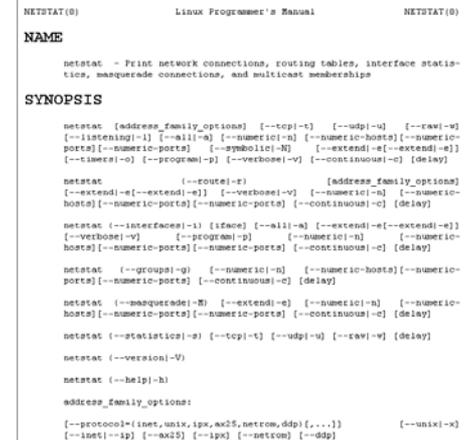
Countermeasures for Backdoors (Windows)

- **TCPView**, available free from www.sysinternals.com provides a GUI-based version, which runs continuously and updates itself when ports are opened and closed:



Countermeasures for Backdoors (UNIX)

- Unix also has netstat:



Countermeasures for Backdoors (UNIX)

- Other tools include:
 - For Linux, *Netfilter/iptables* from www.netfilter.org/ :
 - A free, open source packet filtering tool
 - Built into many Linux distributions
 - For many other UNIX variations, *IPFilter* from www.ipfilter.org :
 - A free, open source packet filtering tool

49

Cross-Network GUIs

- There are a number of tools available which perform a similar function to netcat, except they allow control of a target machine's GUI from a remote machine.
- These tools can be used for useful and legitimate purposes, or for nefarious purposes.

50

Cross-Network GUIs

- Here are some popular Remote GUI Tools:
 - Virtual Network Computing (VNC)
 - <http://www.realvnc.com/>
 - Free, open source tool
 - <http://www.uk.research.att.com/pub/docs/att/tr.98.1.pdf>
 - Runs on many different operating systems
 - Cross platform: Can view a Windows GUI on a UNIX system or vice versa
 - Can use a browser on any machine to connect to a VNC server's built-in web server on port 5800+display number
 - Hackers have modified VNC to use as a remote control backdoor
 - Can be used to 'Shovel a GUI' if firewall blocks incoming connections
 - Microsoft Windows Terminal Services
 - www.microsoft.com/windows2000/technologies/terminal/
 - Microsoft Remote Desktop Service
 - www.microsoft.com/WindowsXP/pro/using/howto/gomobile/remotedesktop/

51

Cross-Network GUIs

- And some more:
 - Citrix MetaFrame
 - www.citrix.com/
 - PCAnywhere
 - www.symantec.com/pcanywhere/
 - Dameware
 - www.dameware.com/
 - GoToMyPC
 - www.gotomypc.com
 - Allows for remote GUI access across the Internet using merely a browser
 - Recently, Citrix acquired Expertcity, Inc, the developer of GoToMyPC

52

Cross-Network GUIs

- And a couple from the 'Computer Underground':
 - Back Orifice 2000
 - Released by the 'Cult of the Dead Cow' hacker group
 - www.bo2k.com
 - SubSeven
 - May be found in the list at <http://packetstormsecurity.org/trojans/>
 - One of the most popular backdoor suites

53

Backdoors Without Ports

- In order to bypass the port restrictions often imposed by firewalls, attackers have moved to using protocols that don't use ports
 - ICMP Backdoors
 - Use the Internet Control Message Protocol (ICMP) as a transmission mechanism
 - ICMP doesn't use ports.
 - For more detail, see:
 - <http://www.networkmagazine.com/article/NMG20000515S0048>
 - Tools that implement this transmission mechanism to provide command shell access include:
 - `Loki` -- <http://www.phrack.org/show.php?p=49&a=6>
 - `007shell` -- <http://www.packetstormsecurity.org/groups/s0ftpj/>

54

Sniffing Backdoors

- Sniffing Backdoors combine:
 - A *sniffer*, which gathers traffic from a LAN,with
 - A *backdoor*, which executes the commands sent in that traffic
- Sniffers grab packets as they pass through the Network Interface Card (NIC) of the computer running the sniffer software. They do not modify these packets in any way.
- By capturing such packets, they can snoop on network traffic, and extract potentially useful (and probably confidential) information.
- For example, see: <http://grc.com/oo/packetsniff.htm>

55

Sniffing Backdoors

- A network card can operate in one of two modes:
 - Non-promiscuous mode
 - Normal mode of operation
 - Accepts packets that are destined only for that machine, based on the MAC address of the NIC.OR:
 - Promiscuous mode
 - Grabs a copy of all packets that pass by the network interface, regardless of their destination.
- Sniffers can set a NIC into non-promiscuous mode or promiscuous mode in order to grab packet traffic either for just the local machine, or for the entire LAN.

56

Sniffing Backdoors

- Some well-known sniffing backdoors, include:
 - Cd00r
 - <http://www.phenoelit.de/stuff/cd00rdescr.html>
 - A non-promiscuous sniffing backdoor
 - Linux-based
 - SAdoor
 - <http://cmn.listprojects.darklab.org/>
 - "A non listening remote shell and execution server"

57

Sniffing Backdoors

- The insidious aspects of sniffing backdoors include:
 - No ports used, so difficult to detect
 - No messages generated from the backdoor; it just copies packets; again, difficult to detect
 - If the sniffer is in promiscuous mode, it may not be on the local machine; it could be anywhere on your LAN, including on the DNS server, etc.

58

Defenses Against Sniffing Backdoors

- Again, harden your system configuration to make it much more difficult for anyone to insert a backdoor on the system(s) in the first place.
- Keep track of the processes running on your systems, especially sensitive ones like firewalls, mail servers, DNS servers, Web servers, etc. If any suspicious processes are active, you need to perform corrective action!
- Run an Intrusion Detection System (IDS). For example:
 - **Snort**
 - <http://www.snort.org/> (free)
 - <http://www.sourcefire.com/> (commercial)
- On some versions of UNIX (but not all), you can detect a promiscuous sniffer by running the **ifconfig** program (not to be confused with Windows' **ipconfig** program) and looking for '**PROMISC**' in its output.
- On Windows, you can run the **Promiscdetect** program available from <http://ntsecurity.nu/toolbox/promiscdetect/>

59

Trojan Horses

- Skoudis defines a Trojan Horse as follows:
A Trojan horse is a program that appears to have some useful or benign purpose, but really masks some hidden malicious functionality

60

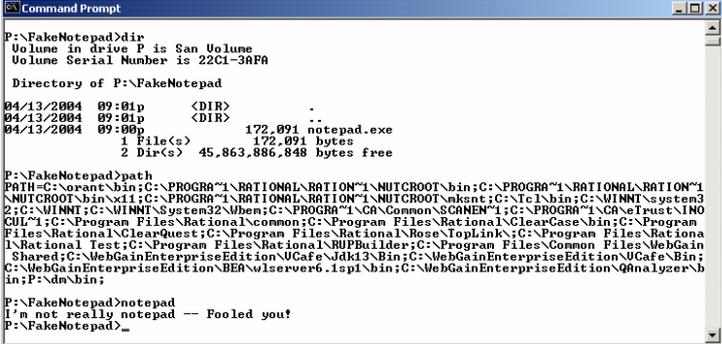
Windows Path Problems

- Here's what's happening:
 - On UNIX and Windows, when you run a program, the system searches the **PATH** for a program of that name, and runs the first one it finds. If it doesn't find one, then you get an error message.
 - There is a major difference between UNIX paths and Windows paths:
 - On Windows, the PATH always (implicitly) includes ".", the current directory
 - On UNIX, it doesn't (and shouldn't)

Why?

65

Windows Path Problems



```
Command Prompt
P:\FakeNotepad>dir
Volume in drive P is San Volume
Volume Serial Number is 22C1-3AFA

Directory of P:\FakeNotepad

04/13/2004 09:01p <DIR>          .
04/13/2004 09:01p <DIR>          ..
04/13/2004 09:00p                172,091 notepad.exe
                1 File(s)      172,091 bytes
                2 Dir(s)      45,863,886,848 bytes free

P:\FakeNotepad>path
PATH=C:\ovant\bin;C:\PROGRAM\NATIONAL\RATION\NUTCROOT\bin;C:\PROGRA\NATIONAL\RATION\NUTCROOT\bin\x11;C:\PROGRA\NATIONAL\RATION\NUTCROOT\sakent;C:\e\bin;C:\WINNI\system32;C:\WINNT\System32\lib;C:\PROGRA\NCA\COMMON\COMMON;C:\PROGRA\NCA\COMMON\Trust\INO\CUL\1;C:\Program Files\Rational\connon;C:\Program Files\Rational\ClearCase\bin;C:\Program Files\Rational\ClearQuest;C:\Program Files\Rational\Rose\TopLink;C:\Program Files\Rational\National_Test;C:\Program Files\Rational\NUPBuilder;C:\Program Files\NehGain\Shared;C:\NehGainEnterpriseEdition\UCafe\jdk\bin;C:\NehGainEnterpriseEdition\UCafe\bin;C:\NehGainEnterpriseEdition\BEA\ulserver6_1spi\bin;C:\NehGainEnterpriseEdition\QAnalyzer\bin;P:\dn\bin;
P:\FakeNotepad>notepad
I'm not really notepad -- Fooled you!
P:\FakeNotepad>
```

66

Program Wrappers

- There are a number of nefarious utilities which can wrap more than one program into a single executable file.
- Attackers can wrap up a benign program together with a malicious one, so that when the user runs the executable, both programs get to run.
 - For more details, see:
 - <http://www.informit.com/articles/article.asp?p=102181&seqNum=2>

67

Steganographic Techniques

- People have figured out ways of hiding information in all kinds of otherwise innocent files:
 - Images
 - Audio files
 - Digital watermarking
- Finally, someone figured out how to do the same thing inside executable programs:
 - Hydan -- <http://www.crazyboy.com/hydan/>

68

Polymorphic Techniques

- Many AntiVirus programs rely on specific signatures for various viruses, worms, etc.
- In response, attackers have developed techniques for changing the code in these viruses dynamically, to avoid producing a fixed signature.
- These guys seem to be incredibly ingenious!
 - I wonder what they could achieve if they applied their ingenuity to useful purposes?

69

Root Kits

- Skoudis defines these as:
RootKits are Trojan horse backdoor tools that modify existing operating system software so that an attacker can keep access to and hide on a machine.
- RootKits can operate at User mode or at Kernel mode for an operating system.

70

UNIX User-Mode RootKits

- On UNIX, User-Mode Rootkits typically are replacement sets of programs that replace the standard operating system programs.
 - They typically look like and act like their legitimate counterparts, except:
 - They have some other nefarious purpose
 - They actively hide their own presence by very ingenious means.

- See:

<http://www.linuxfocus.org/English/November2002/article263.shtml>
<http://packetstormsecurity.nl/UNIX/penetration/rootkits/>

71

Linux RootKit (LRK)

- LRK is actually a family, because it has evolved over time.
- Here is a list of replacements for standard Linux programs provided by the LRK:
 - `login, rshd, chfn, chsh, inetd, passwd, tcpd, sshd, su`
 - `netstat, ps, top, ls, du, ifconfig, syslogd, killall, crontab, pidof, find`

72

Universal Root Kit (URK)

- URK is a general-purpose RootKit that can be applied to a variety of different UNIX implementations
- URK provides the following components:
 - `login`, `sshd`, `ping`, `passwd`, `su`, `pidentd`, `ps`, `top`, `find`, `ls`, `du`, `netstat`, `sniffer`

73

RootKit Prevention on UNIX

- Similar comments apply as before, regarding hardening your system to resist attacks.
- The Bastille Hardening System is available on Linux, HP-UX, and Macintosh OS X
 - <http://www.bastille-linux.org/>
- Other tools are available for other forms of UNIX

74

RootKit Detection on UNIX

- Use a File Integrity Checker, such as Tripwire, AIDE
- Also, there exists a tool called `chkrootkit` at:
 - <http://www.chkrootkit.org/>
- Once you've detected the presence of a RootKit, you basically can't trust anything on that system
 - You have to run tools outside the influence of the infected system's operating system
 - You need trusted programs available on something like a bootable CD-ROM
 - <http://www.stearns.org/staticiso/>
 - <http://fire.dmzs.com>
 - <http://www.knoppix.org/>

75

Windows RootKits

- RootKits have been mostly focused on UNIX, but there are some efforts to produce RootKits for Windows.
- They are not as common, nor as mature as their UNIX counterparts.
- Windows RootKits moved to kernel level almost immediately (<http://www.rootkit.com/>)
- In Windows 2000, and beyond, Microsoft implemented a defense against replacement of system files: [*Windows File Protection \(WFP\)*](#)

76

Windows RootKits

- The Windows Login mechanism can be compromised using standard Windows extensibility mechanisms
 - FakeGINA -- <http://ntsecurity.nu/toolbox/fakegina/>
- Windows File Protection (WFP) can be compromised
 - Code Red II:
 - http://us.mcafee.com/virusInfo/default.asp?id=description&virus_k=99177
 - http://www.cert.org/incident_notes/IN-2001-09.html
 - <http://www.unixwiz.net/techtips/CodeRedII.html>

77

Windows RootKits

- Other techniques used by RootKits on Windows include [***DLL Injection***](#) and [***API Hooking***](#)
- The AFX Windows RootKit uses these techniques
 - http://www.megasecurity.org/trojans/a/aphex/Afx_win_rootkit2003.html

78

Windows RootKit Prevention

- Hardening techniques, again
- Use Windows Security Templates
 - <http://support.microsoft.com/default.aspx?scid=kb:en-us:321679&sd=tech>
 - Consider using the [Win2K Pro Gold Template](#)
- Consider using the free CIS scoring tool available at www.cisecurity.org

79

Windows RootKit Detection

- Consider the use of Fcheck, available from
 - <http://www.geocities.com/fcheck2000/fcheck.html>
- AntiVirus programs can detect RootKits
- Look for unusual port usage, using Fport and TCPView

- Once you've discovered that you have an infected Windows machine, you need to:
 - Back up your system, perhaps for off-line analysis
 - Clean the system completely (wipe disks, etc.)
 - Rebuild your system from scratch, including all patches, etc.
 - Monitor the rebuilt system carefully

80

Kernel-Mode Rootkits

- One level beyond User-Mode RootKits are Kernel-Mode RootKits
- They move to even more sophisticated levels of attack

81

Summary

- Phew! Are you scared yet?
- You probably should be!
- Clearly, there's lots more to this topic, but we only have so much time...

82