
Secure Programming II

(SecProg 2)

Engin Kirda

kirda@eurecom.fr

Administrative Issues

- We have 23 registrations
 - Challenge 1 has started
 - 7 people have solved it so far. Good work
 - You need account information to be able to do the challenges. Send me e-mail if you have not done so
 - Challenge 2 will go online after the buffer overflows lecture, next week

Interesting News From the Field

- Someone who had crawled the social network StudiVZ was arrested
 - The question is: Is it illegal to crawl public information?
 - There is some discussion on this
 - It is against the terms and conditions, but is it against the law?
 - It turned out that the person had tried to blackmail the company (that *is* against the law ;))

General Windows Security

Windows

- 95% of all computers run Windows. Windows is all around you
 - When dealing with security issues, it is important to have knowledge of Windows.
 - Windows is the best example of non-open source system and security issues.
- Windows security is always in the news (major virus, worm and trojan outbreaks in the past, trojans recently were on Windows). Why?
- Seeing the need (finally), Microsoft started a major initiative for security a couple of years ago
 - Attacks are common, e.g., <http://www.windows2000test.com>

Reinventing the wheel?

- Development as non-administrator (“Great” idea ;-))
 - Default configuration on Windows system = admin!
 - Principle of *least privilege*
 - Administrator command shell using **runas.exe** (i.e., su -)
 - Store configuration and user information under `\HKEY_CURRENT_USER`
 - Run services under a restricted user (locking down)
 - Take care in giving debugging privileges
- *I Love You* and *Nimda* would not have worked if computer did not run as **admin**.

Code size (Windows vs. Linux)

- 1992 Windows 3.1 (3M)
- 1995 Windows 95 (15M)
- 1998 Windows NT 4 (20M)
- 1999 Windows 2000 (40M)
- 2000 Red Hat 6.2 (17M)
- 2000 Debian GNU/Linux 2.2 (55M)
 - Linux 2.2 kernel (1.78M)
 - XFree86 3.3.6 (1.27M)
- 2001 Red Hat 7.1 (30M)

Security at Microsoft

- Trustworthy Computing
 - Windows security push
 - Lead for improved security
- What is it?
 - Training, code reviews
 - Threat models and security testing
- SD3 Security Framework
 - Mind setting
 - Principles to adhere strictly

Service Packs and Updates

- Hotfix
 - Single issue / small number of issues
- Security rollup package
 - Single package
 - Multiple hotfixes
- Service pack
 - Major updates
 - Cumulative set of previous updates
 - (optional) Previously *unannounced* fixes
 - (optional) Feature changes
- Major problem: Often **rebooting** is required!

Single User OS (Windows 95/98)

- Almost no security (just like DOS)
 - Anyone can install anything, locking down not possible
- Local Security
 - Highly vulnerable to viruses and trojan horses
 - Highly vulnerable to unauthorized local access/console
 - No file encryption (e.g., like in WinXP).
- Remote Security
 - Highly vulnerable to denial-of-service (weak TCP/IP stack)
 - ping of death, winnuke, land attack
 - If file/print sharing is used
 - Registry can be accessed
 - Win95/98 are not supported by Microsoft anymore (no online updates). There are “zillion” vulnerabilities meanwhile!

Windows 95/98

- Registry
 - used to store system configuration (read/write for all)
- Login Process
 - no authentication – simply press `cancel`
 - determine only profile, don't enforce restrictions
- Profile
 - desktop preferences
 - access to saved passwords (in `.pwl` files)
 - access shared resources, dial-up network
 - Resource Record – Triple `<type, name, passwd>`
 - `passwd` is encrypted with login password

Windows 95/98

- Password files
 - login password is not stored encrypted, instead
 - pwl-file is decrypted with login password and a checksum verified (using user name as well)
 - Windows 95 – algorithm very easy to crack
 - Windows 98 – stronger algorithm (RC4)
 - world-readable
 - vulnerable to brute force / dictionary attacks
 - passwords are always converted to uppercase (makes brute force attacks much easier)
 - unreliable caching mechanism (important information maybe cached)

Windows 95/98 Attacks

- Screen-Saver protection
 - Ctrl-Alt-Del
 - CD-ROM autorun feature to execute programs
 - `autorun.inf` and entry “`open=progname`”
 - Password is stored in Registry
- Malicious Code / Remote exploits
 - 2004 Internet Explorer vulnerabilities (not patched on Win95)
 - Zillion spyware programs, publicly available exploits
 - Good idea not to use Win95/98 – but this is not always possible

Multi User OSs (Windows XP, NT, 2003)

- Obviously – notion of multiple users, multiple tasks
- Authentication
- Access Control
- Privilege Management
- Accounting, Quotas
- Windows NT, XP, 2003
 - object-oriented
 - Systems are based on similar technologies and code-base (i.e., vulnerabilities are usually across multiple platforms)
 - Security Monitor, tightly coupled host and network security

Windows NT Passwords

- NT Maintains backward compatibility to Win95/98, NT passwords can be easy to crack
 - A LANMAN password hash is upper cased, padded to 14 characters, divided into two seven character parts, each of which is used as a key to encrypt a constant.
 - After LANMAN passwd hash is cracked, 2^n (where n is the length of the password) gives the maximum number of case variations that must be tried to get the NT password (about a second ;-)).
 - LANMAN authentication could only be partially disabled (i.e., logging in). The passwd storage scheme and encryption was still weak against brute force attacks.
- In 2001, method was provided to disable LANMAN hashes on 2000 and XP, but not NT

NT and 98 Threat Mitigation Guide

- Syn flooding protection registry hacks
 - Syn flooding is a common attack. Each connection request requires server to allocate certain amount of memory and kernel structures
 - HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\SynAttackProtect(RegDword) = 1
 - HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\TcpMaxHalfOpen(RegDword) = 100 (maximum number of connections)
- Tips and tricks to configure your system (e.g., ICMP redirects, router discovery, RPC stuff)

NT and 98 Threat Mitigation Guide

- NSA has developed a set of recommended permissions for NT servers and workstations
 - “Guide to securing MS Windows NT networks”
- Restrict null-session and anonymous accesses (i.e., to the registry)
- Prevention of the storage of LANMAN passwords
- Patching is recommended by MS, but sometimes this is not possible (i.e., on illegal copies... is this a good idea?)

Windows XP Security Check List

- Provide physical security for the machine ;-)
 - Obvious, but often forgotten.
- Use NTFS on all your partitions (there is “experimental” Linux support)
 - It is faster than FAT32, supports permissions down to the file level
- Disable **Simple File Sharing**.
 - If you are connected to Internet, no firewall, your shares may be readable by practically anyone
 - *Start > My Computer > Folder Options > View > Advanced settings > Simple File Sharing*
 - Does not work on XP Home
 - Make sure your shares are read only and hide them (i.e., using \$)

Windows XP Security Check List

- Use passwords on all accounts
 - Both XP Professional and Home allow user accounts to utilize blank passwords (no more remote logins allowed)
- XP Home Edition **all** user accounts have administrative privileges and no password **by default**.
- Use the Administrator Group with care
- Disable the guest account
 - The guest account has been a huge hacker hole
 - Unfortunately, much easier on Professional than Home. If you disable guest, it is not really “removed”. The only solution is to choose a strong password

Windows XP Security Check List

- Be careful with Internet Connection Sharing (ICS)
 - Anyone maybe able to connect and surf over your network.
Especially interesting for wireless connections ;-)
- Use the Security Configuration Manager
 - SCM tools allow admins to define security templates. These can contain passwd, lockout, audit policies and event log settings, registry values, service startup modes, user rights, permissions, etc.
 - Not available on XP Home edition
- Password security
 - Force policies: GPEDIT.MSC > *Computer Configuration* > *Windows Settings* > *Security Settings* > *Local Policy* > *Security Options*

Spyware

- Any software that monitors and collects information about a user in a covert and unsolicited manner
- Goal of spyware
 - collect sensitive user information and surfing habits
- Task of spyware
 - component must monitor user behavior
 - component must leak information to environment (OS, network)
- Often implemented as browser extensions
 - Internet Explorer Browser Helper Object (BHO)
 - COM object that can hook into Microsoft's Internet Explorer
 - monitor/modify events

Spyware

- Interaction
 - between browser and spyware component
 - COM function invocations (exported by Internet Explorer)
 - between spyware component and operating system
 - Windows API calls
- In addition, it typically has a real company behind it that is making money from the information gathered
 - Adware is any software that injects unsolicited advertisements into a user's workspace
 - Scumware is a specific type of Adware that hides other advertisements with those from its own controlling source

Spyware

Typical routes of infection:

1. spyware is bundled with legitimate software package
 - end-user license agreement (EULA) even informs about this fact
 - EULA is very long (often hundreds of pages), user accepts
 - classic examples are shareware programs
 - P2P file-sharing clients (e.g., Kazaa)

2. “drive-by” downloads
 - exploit browser bug, in particular, vulnerabilities of Internet Explorer
 - WMF (Windows meta file) exploit, around Christmas 2005
 - arbitrary code execution via mismatched DOM objects (December 2005)
 - insufficient ActiveX security settings

3. fake dialogs
 - display “Would you like to optimize your Internet” and perform installation when user agrees

Spyware

- Spyware is becoming a major security issue
 - Analysis performed by Webroot and Earthlink showed that a large portion of Internet-connected computers are infected with spyware (... Windows problem)
 - Spyware tends to monitor the behavior of users and steal private information (profit, targeted advertisement, etc.)
 - Antispyware programs: Booming business... but how effective?
 - Just like virus/worm detectors, they use signatures
 - Malware needs to be *known*

Spyware

- Spyware authors have many options when it comes to looking for good vantage points
 - Layered Service Providers (LSPs) sit between Winsock and the Base Service Provider
 - E.g., filter network traffic, intercept user data / actions, etc.
 - Browser Helper Objects (BHOs) (i.e., plug-ins) and Toolbars for IE seem to be the most popular spyware implementation techniques
 - A study showed that of 120 samples, 90 had BHO architecture

Component Object Model

- COM is a binary standard realized by Microsoft to enable a component-based market
 - Every COM object is derived from a set of interfaces
 - All COM interfaces have as their root interface *IUnknown*
 - *IUnknown* contains a function called *QueryInterface()*
 - Using this function, one can query for implemented interfaces and get a pointer to them
 - *QueryInterface()* enables the discovery of capabilities

Browser Helper Objects (BHOs)

- A BHO is in essence...
 - ... a simple Component Object Model (COM) object that implements the *IObjectWithSite* interface.
 - IE will load all registered BHOs (that are COM servers) when it starts. It does this by looking at the Class Identifiers (CLSIDs) under
 - \HKLM\SOFTWARE\Windows\CurrentVersion\Explorer\Browser Helper Objects
 - The *IObjectWithSite* interface has a function called *SetSite()*
 - When IE is started, it instantiates the BHO and calls *SetSite* with a pointer to itself.
 - BHO has access to functions and pointers in IE (e.g., open a new window, etc.)

Browser Helper Objects (BHOs)

- A BHO can “listen” to events fired by IE such as *Before Navigate*, *Navigate Complete*, *New Window*, etc.
 - The events of interest are defined in the interface *IWebBrowser2* (check the MS documentation)
 - Our previous research dealt with a novel detection technique: We “imitate” IE and try to find out what the spyware is doing
 - We generate events and statically and dynamically look at suspicious WinAPI calls. First results are encouraging
 - USENIX Security 2006, Vancouver, Canada

Useful Spyware Tools

- HijackThis (www.hijackthis.de)
 - Low-level tool, very useful in doing research as well as removal
- Spybot, Adware: Freeware tools
 - Signature based so they do not catch all spyware
 - We currently have a project where we crawl the web and test how effective signature-based solutions are

Windows XP Service Pack 2

- A set of security “technologies” for improvement of situation
 - Network protection
 - Improved firewall, reduced RPC attack surface (reduced credentials)
 - Memory protection
 - MS version of StackGuard has been deployed (/GS switch)
 - E-mail handling
 - Outlook has been “fixed”, recompiled
 - Web browsing security
 - Privileges / locking down
 - Automatic updates (there were reports of problems at first)

Was Windows XP SP2 more secure?

- “It is late, it is large, but a step in the right direction”
However...
 - No execute setting is not present in current hardware architecture of most 64-bit and 32-bit processors on the market. Features not useful yet.
 - Will take years until new hardware and software changes trickle down to the masses ☹
 - Virus writers will not give up and will keep busy.
- Enjoy XP SP2 firewall with care: Only *inbound* connections are checked
 - It is possible for code to modify setting and firewall (e.g., Phrack articles). See recent postings.

Windows Vista Security

- The “wow!” effect ;-)
 - Vista == You take Mac OS UI gimmicks and combine them with new features
- Now, most applications run in non-admin mode
 - User account control: If privilege is required, Secure Desktop mode is activated
- Bitlocker Drive encryption, Encryption FS
 - Full volume encryption, key can be stored on USB

Windows Vista Security

- Windows Firewall
 - Has been improved (e.g., outbound connections, port ranges, IP ranges, etc.)
- Windows Defender
 - Microsoft's anti-spyware utility is included
- Windows Parental Controls
 - Web content blocking, time limitations on account, restrictions on programs executed, etc.
- Exploit prevention
 - Address Space Layout Randomization (ASLR)
 - Encryption of function pointers
 - Stack overflow detection (canary mechanism)

Windows Vista Security

- Data Execution Prevention (DEP)
 - Vista supports full support for NX feature of processor
 - Problem: Not all applications / processors are DEP-aware
- Application isolation
 - Mandatory Integrity Control
 - Application in a lower integrity level cannot access resources in higher integrity level
- Network Access Protection (NAP)
 - Computers should conform to preset “system health” level, otherwise, network access limited or denied (e.g., updates need to be installed)

Windows Vista Security

- Process Isolation
 - Previous versions of Windows, all services ran under same session
 - Not so anymore: Isolation Session 0
 - Normal process can not show popups or dialogs anymore
 - If it does, it will be invisible and will sit in the background
 - To interact, processes need to use Windows calls (so there is stricter control)

Windows Vista Security

- File and registry virtualization
 - Windows programmers generally assumed that they are admin
 - Thousands of programs exist out there so backwards compatibility is important
 - However, all-access registry operations have been disabled
 - Microsoft has introduced a file and registry virtualization for backwards-compatibility
 - Application writes to a “per user” location, does not realize it, it is transparent

.NET Framework Security

- Managed execution and type safety
 - Exception manager
 - Buffer overflows not possible
 - Security Engine
 - Code Access Security
 - But wait... there is “unmanaged” mode...
- CLR Integrated Security
 - Code access security
 - Role-based security
- .NET Framework Libraries
 - Cryptography
 - Web Services and Applications

.NET Framework Security

- Remote code:
 - With the growth of the Internet, applications are increasingly downloaded from remote sources
 - Users are susceptible to executing malicious code
- The proposed solution by Microsoft:
 - Introduced the .NET framework, where machine-independent byte-code is executed on a virtual machine
 - .NET is an implementation of the Common Language Infrastructure (CLI)
 - Consists of Common Type System (CTS)
 - .NET is type-safe and memory-safe

.NET Framework Security

- An important feature of .NET
 - It allows access to native libraries (i.e., legacy code support)
 - .NET applications are called *managed* and native code is referred to as *unmanaged* code
 - The runtime environment can enforce security restrictions by relying on type and memory-safety
 - Security model is called Code Access Security (CAS)
 - CAS uses *evidence* provided by the program and security policies to generate permissions (e.g., file access)
- Unfortunately, the execution of unmanaged native code is not restricted by the security model
 - Hence, an attacker can completely circumvent the .NET security mechanisms

Invoking Unmanaged Code in .NET

- To support interoperability with languages such as C, C++
 - CLI uses a mechanism called *Platform Invoke Service* (P/Invoke)
 - Because native code can modify the security state of user's environment, .NET permissions are *full trust*
 - The native code is run within the same process as CLI, an attacker could modify .NET runtime itself
 - Microsoft suggests P/Invoke to be used for highly-trusted code
 - This, however, cannot always be feasible

Conclusion

- Today, we looked at Windows security
- Next week, we start looking at basics (e.g., assembler, calling conventions, etc.) to be able to understand security issues such as stack overflows
- See you next week ;)