Binary Analysis: Finding Secrets in Code

Nish Bhalla
Founder, Security Compass
Agenda

Product Review Methodology

Basics

Debugging Steps

Ollydbg

Process Isolation

isDebuggerPresent

ISAPI Analysis
Typical Product Review Methodology

- **What it is:** Product reviews examine vendor applications using a black-box approach to find security vulnerabilities.

- **Why:** Product reviews discover vulnerabilities before the product is shipped, thereby decreasing potential security updates, enhancing your organization's reputation amongst customers and creating a competitive advantage.

- **How:** Product reviews examine applications to determine security vulnerabilities by performing the following steps:
  
  1. **Threat Analysis**- Incorporate the full threat analysis methodology.
  2. **Design Analysis**- Map out the application use and functionality requirements.
  3. **Architecture Review**- Examine the dataflow model and the controls “by design” on each communication point to secure data.
  4. **Application Penetration Testing**- Perform black and grey-box testing on the product on a variety of potential end-user environments.
Threat Analysis

• What is Threat Analysis?

• Threat Analysis or threat modeling is the process of systematically deriving the **key threats** relevant to an application in order to efficiently **identify** and **mitigate** potential **security weaknesses** before deployment.

• It is a method to determine the unique threats that an application might face; it is a systematic method of finding security issues in an application by **forcing developers to think like an attacker**.

• Security staff can **focus** their resources on the most important issues an application faces after performing this activity.
Design And Architecture Review

- Design Review: Understand what the thought process was when the application was originally designed
- More often than not the design and the end product are two completely different applications
- Architecture Review: Evaluate the deployed application’s architecture and implementation in a real world scenario

??? SECURE BY DEFAULT ???
• **Our Sample Application** is an ISAPI. It requires a pass phrase to browse into the website

• Why did I choose an ISAPI and not a stand alone application?
  
  - Developers often store credentials of back end databases in ISAPIs
  - ISAPIs are commonly where algorithms are stored
  - ISAPIs are typically not expected to land in the hands of a hacker
  - ISAPIs are typically reviewed for lost code and lost developers
  - ISAPIs can help you crash systems that are otherwise secure
  - I recently reviewed a few ISAPIs and I am partially basing this example on observations made in those reviews
Agenda

Product Testing Methodology

Basics

Debugging Steps

Ollydbg

Process Isolation

isDebuggerPresent

ISAPI Analysis
Discovering Additional Derogatory Netscape References

A Web Exclusive from Windows IT Pro
Editors
IT News
InstantDoc #6965
Windows IT Pro

Subscribe to Windows IT Pro | See More Security Articles Here | Reprint

The "Netscape engineers are weenies" reference in Microsoft Visual InterDev 1.0 earlier this month might not just be an isolated incident. In Microsoft Security Bulletin MS01-025, Microsoft mentions only Visual InterDev 1.0 and this associated the phrase "as containing the now famous strings in reverse-chars order. In other words, the reversed phrase is still there!" However, a reader's sharp eye has led to the discovery that the reference appears in all sorts of other DLLs that install with Visual InterDev 6.0, Visual Studio 6.0, and Visual Studio 97. The same phrase has turned up in the dynamically linked libraries msvc7.dll and mscorlib.dll.

According to the Microsoft Developer Network (MSDN) DLL Help database, mscorlib.dll is also part of Visual InterDev's LinkView function—the same function to which the previously reported lounge.dll contributed. Mscorlib.dll is also a part of the much older Visual Studio 97. Microsoft is aware of this new discovery and is currently investigating it.
This is an eval for strlen function

Please enter the secret to pass the gate

OK

Lets attempt HackInTheBox as the key

Application Background: An ISAPI that expects a valid string to allow you to pass the “Gate” to get to the application.
Maybe it is hard coded.

HackInTheBox didn’t work, but we got a string back “You don’t have a valid key …”
Agenda

Product Testing Methodology

Basics

Debugging Steps

Ollydbg

Process Isolation

isDebuggerPresent

ISAPI Analysis
ISAPI – Debugging Steps

- Attach/Load ISAPI in debugger
  - Decide which debugger.

- Bypass any anti-debugging steps

- Step through binary

- Set breakpoints at best locations possible
ISAPI – Debugger, Ollydbg

- Lets attach a debugger to the DLL to see what is going on.
  - Ollydbg
  - GDB / DDD [GUID GDB]
  - Windbg

- Ollydbg is one of the best User Mode debuggers available and it is free

- To learn more about ollydbg, download and read the help files. Additional links will be provided at the end of the presentation.

- Ollydbg can be downloaded from [http://www.ollydbg.de/](http://www.ollydbg.de/)

- Ollydbg has a community forum which has moved to this [http://www.asmcommunity.net/board/](http://www.asmcommunity.net/board/). The older messages will be available sometime in the near future
Agenda

Product Testing Methodology

Basics

Debugging Steps

Ollydbg

Process Isolation

isDebuggerPresent

ISAPI Analysis
• Attach inetinfo.exe process.  [Demo]
• The default in OllyDbg breaks at WinMain. We shall wait for it to pause at that location.
ISAPI - Ollydbg

- View Executable Modules.
- The ISAPI can’t be found 😞
- What should be our next step?
- Ideas / Suggestions?
- Search Google / Microsoft what is happening behind the scenes?
Agenda

Product Testing Methodology

Basics

Debugging Steps

Ollydbg

Process Isolation

isDebuggerPresent

ISAPI Analysis
IIS Process Isolation - Microsoft

- Application Protection or Isolating Applications refers to the process in which applications (ISAPIs) are run. It means configuring them to run in a process (memory space) that is separate from the Web server and other applications. You can configure applications to have one of three levels of application protection:
  - Low (in-process) application protection.
  - Medium (pooled) application protection.
  - High (isolated) application protection.

- IIS 5.0 & 6.0 offers three levels of application protection.
  - Low - ISAPI is inside inetinfo.exe process space
  - Medium – ISAPI is in dllhost.exe process space
  - High – ISAPI is in dllhost.exe process space.

- In case you encounter a In IIS 4.0 based application, the ISAPI ran in either (Inetinfo.exe) or in a process separate from Web services (DLLHost.exe), Low & High only.
ISAPI – Ollydbg

- Easiest for debugging is make it low; in-process application protection. We might have other applications running in Dllhost.exe process space and we don’t want to kill our other apps.

- Restart IIS

- Attach Inetinfo.exe and let the application loop through while you input data into the HTML form

- Should be able to find Secret.dll in loaded executables (View / Executable Modules) in inetinfo.exe

- Application is Terminated

- IDEAS / Suggestions?
Agenda

Product Testing Methodology

Basics

Debugging Steps

Ollydbg

Process Isolation

isDebuggerPresent

ISAPI Analysis
ISAPI – isDebuggerPresent()

- ntdll.ZwTerminateProcess

- Viewing names in Secret.dll we find that there is a call to function isDebuggerPresent.

- The IsDebuggerPresent function terminates execution of the process. This is a common technique used to discourage debugging of applications.
Manually - Bypassing iSDriverPresent

**Method 1**

- Set a breakpoint on the `IsDebuggerPresent` function.
  - we’ll load the command-line plug-in (Alt+F1)
  - `bp IsDebuggerPresent`

- Once the break point is reached, executes “Step Into” twice (Shift+F7 * 2) and stop

- By right clicking in the Disassembler pane and selecting Follow in Dump | Memory Address, the location and value of the `IsDebuggerPresent` function is displayed in the Dump pane. The location is 7FFDA002

- **01 00 FF FF FF FF 00 00 40 00 A0 1E 19 00**

- Right-clicking the first value in this string (01) and selecting “Binary\Fill with 00’s”
Manually - Bypassing iSDdebuggerPresent

Method 2

- A simpler way to do this is to load Ollydbg’s command-line plug-in (Alt-F1).

- Insert the following command
  set byte ptr ds:[fs:[30]+2] = 0

- The command changes the return value of the API (isDebuggerPresent) to always be 0.
Plugins - Bypassing isDebuggerPresent

- **Hide Debugger**
  - Hides debugger from the Kernel32.isDebuggerPresent function

- **isDebuggerPresent Plugin**
  - Hide mode, hides debugger from the Kernel32.isDebuggerPresent function
  - Extra Hide
  - Dump process (need to provide from where to where).
Plugins - Bypassing isDebuggerPresent

- Olly Invisible
  - System Wide Hooking - This method Ollydbg will be hidden to the whole system.
  - User Wide Hooking - This method Ollydbg will be hidden to the current user session only.
  - Only Target Process – This method hides to targe process only (saves hooking)

- Functions Olly Invisible hides
  - IsDebuggerPresent
  - IsBeingDebugged
  - CsrGetProcessId
  - ZwQuerySystemInformation
  - ZwQueryInformationProcess
Agenda

Product Testing Methodology

Basics

Debugging Steps

Ollydbg

Process Isolation

isDebuggerPresent

ISAPI Analysis
ISAPI - Analysis

-Enumerate functions, imported and exported by ISAPI
-Enumerate strings inside ISAPI
-Review code of key functions
-Set breakpoint on any or all key aspects
Enumerate functions, imported and exported by ISAPI

- The functions can be enumerated using Quick View or Quick View Plus utility

- Same can be done using dumpbin utility, provided with Visual Studio
dumpbin /EXPORTS secret.dll

- In Ollydbg, to view the list of functions, select View Executable Modules, right-clicking on Secret.dll option from the list and selecting view names displays the functions being imported and exported
Enumerating strings

- All the strings being used inside secret.dll can be viewed either by using the “strings” utility available from sysinternals website.

- ASCII strings inside secret.dll can be viewed by right-clicking inside the Disassembler pane where secret.dll is loaded and selecting Search for | All referenced Text strings.

- Set a breakpoint on any interesting strings that are seen. Example: “You don’t have a valid key, The key you attempted was”. The error we had seen on attempt to login.
ISAPI - Analysis

Review code of key functions

- Key Functions such as strcpy and strcat are being used

- To select references on import, right click on the function and select references on import. A new pane with a list of references pops up. Set a breakpoint on the references
ISAPI - Analysis

- Browse to the website and provide a string.
  - String providing strategy
    - AAAAAAAAAAAAAAAAAAA
    - AAAABBBBCCCCDDDDD
    - 1111222233334444

- The application should stop before the Failed secret error message.

- Tracing the call in code a few lines above the breakpoint, we note a comparison is performed between the data input and a string locally stored in the binary, “SecurityCompass”.

- GAME OVER !!
ISAPI - Input Valid String

- http://localhost/ and providing the string SecurityCompass.

- To obtain the dll either browse to securitycompass.com and obtain it from the resources section or by browsing to http://www.webhackingexposed.com/.
  Available only after July 01, 2006.

You may enter any time, The key you attempted was SecurityCompass
Some Interesting Links

- OllyDebugger Web Site
  http://www.ollydbg.de/

- OllyDebugger Forum for Discussion
  http://www.asmcommunity.net/board/

- Open Reverse Code Engineering Community Website
  http://www.openrce.org/

- Site with a lot of interesting tools
  http://www.sysinternals.com/Utilities/Strings.html

- Assembly Tutorials
  http://spiff.tripnet.se/~iczelion/tutorials.html
Security Compass Profile

• Our consultants have serviced large (Fortune 500) and medium sized companies across most major industries
• We have worked for major security players, including Foundstone and Deloitte
• We have co-authored or contributed to several security books, including:
  − Buffer Overflow Attacks: Detect, Exploit & Prevent
  − Windows XP Professional Security
  − HackNotes: Network Security
  − Writing Security Tools and Exploits
  − Hacking Exposed: Web Applications, 2nd Edition
• We have presented at and continue to present at security conferences, including:
  − Reverse Engineering Conference 2005 in Montreal; HackInTheBox 2005 in Malaysia; ISC2's Infosec Conferences in Las Vegas, NYC, Toronto & DC; CSI NetSec; DallasCon; ToorCon; and Freenix.
• We present and contribute to open source projects:
  − Chair at OWASP Toronto, Presented at OWASP Toronto, Contributed to YASSP Project (Lead by SANS and Xerox), Botan Crypto library, Cutlas P2P network & VNCCrack
QUESTIONS?

- Contact:
  Nish Bhalla ([Nish@securitycompass.com](mailto:Nish@securitycompass.com))
  Founder, Security Compass
  Toronto, Ontario, CANADA: 647.722.4883
  Shrewsbury, New Jersey, USA: 201.390.9198