Skeletons in Microsoft's Closet
- Silently Fixed Vulnerabilities

Andre Protas
Steve Manzuik
- Introductions / Outline
  - That’s this slide so we are done with that.

- Non-Disclosure
  - Politics of not disclosing a vulnerability when it is fixed.
  - What vendors practice this?
  - Why is this bad?

- Silently Fixed Bug Hunting
  - Our methodology.
  - Tools we used.
  - Reversing 101.

- Potential Hits
  - Output from the tools.
  - Identifying potential issues.
Presentation Outline

- Filtering the List
  - What we ignored.
  - Why we ignored?
- Vulnerabilities Found
  - Issues identified
  - Vulnerabilities vs. “Security Enhancements”
- Details
  - Security enhancements
  - Vulnerabilities found
- Vulnerability Exposed
  - Where we found the vulnerability
  - How do you exploit the vulnerability
- Demo
- Questions
  - Buy us beer!!!
Why This Topic?

ASN.1 Story

- Vendor released a patch (MS04-007) fixing what appeared to be 1 issue
- Vendor who discovered the vulnerability had 2 advisories
- Exploit code for a third issue was created and sold privately on the vulnerability market.

- Upon further analysis a total of seven issues were actually fixed.
- Apparently 1 = 7. New math?
- So who cares? Or better yet why care?
NonDisclosure

What is Non-Disclosure?
- In the context of Information Security, Non-Disclosure is the act of not disclosing any details of a security vulnerability.
- Many vendors, not just Microsoft, practice this on a regular basis.

Politics
- Disclosure = Press (bad?)
- Does press affect buying habits?
- Perception of security outweighs reality.
Non Disclosure

Vendor View
- Why disclosure internally found issues
- NDA agreements with third party consultants
- Adds release process overhead
- Customers install all patches anyways. Right?

“If a vulnerability is found in a component, you should look for all related issues in that component”

- Writing Secure Code Second Edition
  Microsoft Press
Why This Is Bad

- Customers do *NOT* install all patches.
- Affects patch management methodology.
- Signature based vendors may not catch on.

A Word About Signatures

- Many vendors do not have the resources or skill to reverse a patch.
- The nature of our industry means that being first is best.
- Being first doesn’t always equal being right.
Silently Fixed Bug Hunting

Methodology
- Identify patches that most likely have silent fixes
- Document publicly known issues addressed in the patch
- Catalog files in the patch
- Gather pre-patch files that are related
- Compare prepatch.dll with patch.dll
- Identify areas of interest
- Review interest areas for potentially exploitable flaws
- Exploit flaw in pre-patch environment
- Test exploit against post-patch environment
- Test against a signature based security solution
Silently Fixed **Bug Hunting**

**Patch Identification**

- Priority system for reviewing patches for silently fixed vulnerabilities.
  - Anonymous remotely accessible patched system functionality.
  - Non-Anonymous remote system functionality.
  - Non-remote system functionality
Silently Fixed Bug Hunting

Document publicly known issues addressed in the patch

- During Patch Tuesday, details may be at a minimum.
  - Use CVE and Advisory details (if they exist) to try to pinpoint the disclosed vulnerability.
  - Monitor exploit posts for the vulnerability to better understand the function that is being exploited.

"The MIDL_user_allocate function in the Microsoft Distributed Transaction Coordinator (MSDTC) proxy (MSDTCPRX.DLL) allocates a 4K page of memory regardless of the required size, which allows attackers to overwrite arbitrary memory locations using an incorrect size value that is provided to the NdrAllocate function, which writes management data to memory outside of the allocated buffer. "

---

![CVE-2005-2119](image-url)
Silently Fixed **Bug Hunting**

**Catalog Files In The Patch**

- Usually pretty easy using the `/x` command on the installer.
- Filter update installer files out of the directory, and only include the files that were updated as part of the patch itself.

<table>
<thead>
<tr>
<th>File Name</th>
<th>Version</th>
<th>Date</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caterv.dll</td>
<td>2002.2</td>
<td>3529.0</td>
<td>05-Sep-2005</td>
</tr>
<tr>
<td>Caterv.dll</td>
<td>2002.2</td>
<td>3529.0</td>
<td>05-Sep-2005</td>
</tr>
<tr>
<td>Cluctate.dll</td>
<td>2002.2</td>
<td>3529.0</td>
<td>05-Sep-2005</td>
</tr>
<tr>
<td>Cluctate.dll</td>
<td>2002.2</td>
<td>3529.0</td>
<td>05-Sep-2005</td>
</tr>
<tr>
<td>Comact.dll</td>
<td>2002.2</td>
<td>3529.0</td>
<td>05-Sep-2005</td>
</tr>
<tr>
<td>Comact.dll</td>
<td>2002.2</td>
<td>3529.0</td>
<td>05-Sep-2005</td>
</tr>
<tr>
<td>Comact.dll</td>
<td>2002.2</td>
<td>3529.0</td>
<td>05-Sep-2005</td>
</tr>
<tr>
<td>Comact.dll</td>
<td>2002.2</td>
<td>3421.3529</td>
<td>05-Sep-2005</td>
</tr>
<tr>
<td>Comact.dll</td>
<td>2002.2</td>
<td>3529.0</td>
<td>05-Sep-2005</td>
</tr>
<tr>
<td>Comact.dll</td>
<td>2002.2</td>
<td>3529.0</td>
<td>05-Sep-2005</td>
</tr>
<tr>
<td>Dircopy.exe</td>
<td>2002.2</td>
<td>3529.0</td>
<td>26-Jun-2005</td>
</tr>
<tr>
<td>Ext.dll</td>
<td>2002.2</td>
<td>3529.0</td>
<td>05-Sep-2005</td>
</tr>
<tr>
<td>Medicate.dll</td>
<td>2002.2</td>
<td>3529.0</td>
<td>05-Sep-2005</td>
</tr>
<tr>
<td>Medicate.dll</td>
<td>2002.2</td>
<td>3529.0</td>
<td>05-Sep-2005</td>
</tr>
<tr>
<td>Mediate.dll</td>
<td>2002.2</td>
<td>3529.0</td>
<td>05-Sep-2005</td>
</tr>
<tr>
<td>Mediate.dll</td>
<td>2002.2</td>
<td>3529.0</td>
<td>05-Sep-2005</td>
</tr>
<tr>
<td>Mediate.exe</td>
<td>2002.2</td>
<td>3529.0</td>
<td>30-Aug-2005</td>
</tr>
<tr>
<td>Mocch.dll</td>
<td>2002.2</td>
<td>3529.0</td>
<td>05-Sep-2005</td>
</tr>
<tr>
<td>Mode.dll</td>
<td>2002.2</td>
<td>3529.0</td>
<td>05-Sep-2005</td>
</tr>
<tr>
<td>Mode.dll</td>
<td>2002.2</td>
<td>3529.0</td>
<td>05-Sep-2005</td>
</tr>
<tr>
<td>Mode.dll</td>
<td>2002.2</td>
<td>3529.0</td>
<td>05-Sep-2005</td>
</tr>
<tr>
<td>Ole32.dll</td>
<td>5.0.2195.7059</td>
<td>05-Sep-2005</td>
<td>957,712</td>
</tr>
<tr>
<td>Ole32.dll</td>
<td>5.0.2195.7059</td>
<td>05-Sep-2005</td>
<td>69,392</td>
</tr>
<tr>
<td>Ole32.dll</td>
<td>5.0.2195.7059</td>
<td>05-Sep-2005</td>
<td>35,624</td>
</tr>
<tr>
<td>REnv.dll</td>
<td>5.0.2195.7059</td>
<td>11-Mar-2004</td>
<td>449,908</td>
</tr>
<tr>
<td>REnv.dll</td>
<td>5.0.2195.7059</td>
<td>05-Sep-2005</td>
<td>212,240</td>
</tr>
<tr>
<td>Ssp.dll</td>
<td>5.0.2195.7040</td>
<td>21-Apr-2005</td>
<td>6,369,036</td>
</tr>
<tr>
<td>Ssp.dll</td>
<td>5.0.2195.7040</td>
<td>09-Sep-2005</td>
<td>73,440</td>
</tr>
<tr>
<td>T-Fast.dll</td>
<td>2002.2</td>
<td>3529.0</td>
<td>05-Sep-2005</td>
</tr>
<tr>
<td>Votrip.dll</td>
<td>2002.2</td>
<td>3529.0</td>
<td>05-Sep-2005</td>
</tr>
</tbody>
</table>
Silently Fixed Bug Hunting

Gather pre-patch files that are related

- VMWare images can be a huge help here.
  - Keep VMWare images of each SP/UR as well as a current one for use against upcoming patches
- Keep a solid filing convention for the files that are to be analyzed to avoid confusing
  - Especially useful in batch analysis of service packs or update rollups
Silently Fixed **Bug Hunting**

**Diffing In General**

- The process of enumerating the changes made between two entities
  - Typically performed on files to look for textual differences (+ / - / change)
  - Great for learning the differences between configuration files

- But this can also be used in binary files.
  - Enumerate the functionality between two dlls/exe/etc files (+ / - / change)
  - Great for learning what security/functionality enhancements may have been introduced in the patch.
  - Use IDA Pro to reverse engineer the system file both pre- and post-patch.
Silently Fixed **Bug Hunting**

**Reverse Engineering 101**

- A compiled file can be disassembled to show the machine code being processed for that file.

- Allows for pseudo-translation into source code.

- A disassembly can be used to find flaws, hidden APIs, or any other number of low-level functionality that may/may not be documented in standard references.

- Our Use? We use reverse engineering to dissect the security enhancements applied in Microsoft patches.
Silently Fixed **Bug Hunting**

**Tools we used**

- File information extraction
  - Muddle diff
  - Strings diff
  - Symbol retrieval
- Analysis with IDAPro
  - Custom IDA(Python)? Plugins
  - Sabre BinDiff
- Normal Debugging/Testing Environments
  - Pre/Post Patch VMWares (not ‘snapshots’)

---

*eEye Digital Security®*
What About Service Packs / Update Rollups?

Automation is key

- Enter BS (binary_diffing starter)
- Useful suite of scripts to automate the basic binary diffing dependencies to allow for less wasted time.
  - Informational Gathering (sizes, names, versions, md5s, etc)
  - PDB symbol retrieval
  - IDB generation (pre/post PDB symbols)
  - Muddle/String diff utility
  - Allows for a specified IDC script to be run
- This tool allows for a complete basic reconnaissance of a service pack / update rollup once it has finished analyzing all of the files (pre and post patch) that were update.
Potential Hits

Output – IDA Split Screen
Potential Hits

Output – Strings Diff

REMOVED IN THE LATEST BUILD

______________________________
_NDBSERVERCALL2
_ELFDEREgistereventSource
__IElf_handle_pendown
_ELFFACLOSIEl
_FIXCONTEXTHANDLESFORRECOR

END OF OLD REMOVED

ADDED IN THE LATEST BUILD

______________________________
__iMP__I_RPCBINDINGISCLIENTLOCAL
_STRINGCOPYWorkRM
__FD:CONTEXTHANDLESFORRECOR
__FPNRPCSESSIONSTRICKCONTEXTHANDLE

END OF ADDED
Potential Hits

Output – BinDiff
Filtering the List

Ignored Results

- Although we didn’t ignore tool output completely, some information was not used as much except for a support role to the IDA twin-disassembly or the BinDiff IDA plugin.
- Strings_Diff
- Removed subroutines from disassembly.
- Muddle_Diff
Filtering the List

Why we ignored

- We ignored much of the string diff generation, as there were many false positives that were reported by the string generation tool.

- Although removed functionality could be interesting as well, we were primary concerned with the added functionality / security enhancements.

- Most of the muddle output was ignored as muddle can generate many false positives (complicated data structures could be equal, but would alert a differ tool as changed).
Security Enhancements Found

Not Vulnerabilities But...

- The release notes from Windows 2000 Update Rollup 1 says;

  “This update rollup contains security-related updates that were produced for Windows 2000 between the release of Windows 2000 SP4 and April 30, 2005. On April 30, 2005, the contents of Update Rollup 1 were locked for final testing by Microsoft and customer beta testing. This update rollup also contains several important non-security updates. This article contains detailed information about this update rollup, answers frequently asked questions, and lists the updates that are included in this update rollup. “

- Do you understand this paragraph?
Security Enhancements Found

Not Vulnerabilities But...

- Non-Strict RPC connections now enforced
  - Previously allowed for context switching between RPC interfaces within the same process (i.e. services.exe)
  - Allows for RPC evasion (via ALTER_CONTEXT)
  - Potential DoS (access violation) from improperly checked context handle from possibly

Example: Eventlog.dll
CLSID: 82273fdc-e32a-18c3-3f78-827929dc23ea

NOTE: This is the ONLY dll with this change.
Vulnerabilities Found

Don’t worry. We were not done looking for silently fixed bugs.

<table>
<thead>
<tr>
<th>File Name</th>
<th>Version</th>
<th>Date</th>
<th>Time</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cetmon.dll</td>
<td>2.0.5029.0</td>
<td>09-Sep-2005</td>
<td>00:10</td>
<td>160,649</td>
</tr>
<tr>
<td>Cetmon.dll</td>
<td>2.0.5029.0</td>
<td>09-Sep-2005</td>
<td>00:10</td>
<td>595,729</td>
</tr>
<tr>
<td>Olscep.dll</td>
<td>2.0.5029.0</td>
<td>09-Sep-2005</td>
<td>00:10</td>
<td>97,040</td>
</tr>
<tr>
<td>Olscep.dll</td>
<td>2.0.5029.0</td>
<td>09-Sep-2005</td>
<td>00:10</td>
<td>553,104</td>
</tr>
<tr>
<td>Collot.dll</td>
<td>2.0.5029.0</td>
<td>09-Sep-2005</td>
<td>00:10</td>
<td>42,744</td>
</tr>
<tr>
<td>Collot.dll</td>
<td>2.0.5029.0</td>
<td>09-Sep-2005</td>
<td>00:10</td>
<td>107,904</td>
</tr>
<tr>
<td>Consadmin.dll</td>
<td>2.0.5029.0</td>
<td>09-Sep-2005</td>
<td>00:10</td>
<td>97,040</td>
</tr>
<tr>
<td>Consadmin.dll</td>
<td>2.0.5029.0</td>
<td>09-Sep-2005</td>
<td>00:10</td>
<td>342,288</td>
</tr>
<tr>
<td>Consadmin.dll</td>
<td>2.0.5029.0</td>
<td>09-Sep-2005</td>
<td>00:10</td>
<td>1,471,268</td>
</tr>
<tr>
<td>Consadmin.dll</td>
<td>2.0.5029.0</td>
<td>09-Sep-2005</td>
<td>00:10</td>
<td>628,396</td>
</tr>
<tr>
<td>D��eepol.exe</td>
<td>2.0.5029.0</td>
<td>09-Sep-2005</td>
<td>00:10</td>
<td>1,833,908</td>
</tr>
<tr>
<td>Exe.dll</td>
<td>2.0.5029.0</td>
<td>09-Sep-2005</td>
<td>00:10</td>
<td>342,440</td>
</tr>
<tr>
<td>Modlog.dll</td>
<td>2.0.5029.0</td>
<td>09-Sep-2005</td>
<td>00:10</td>
<td>96,016</td>
</tr>
<tr>
<td>Modlog.dll</td>
<td>2.0.5029.0</td>
<td>09-Sep-2005</td>
<td>00:10</td>
<td>726,208</td>
</tr>
<tr>
<td>Modlog.dll</td>
<td>2.0.5029.0</td>
<td>09-Sep-2005</td>
<td>00:10</td>
<td>1,206,400</td>
</tr>
<tr>
<td>Modlog.dll</td>
<td>2.0.5029.0</td>
<td>09-Sep-2005</td>
<td>00:10</td>
<td>153,872</td>
</tr>
<tr>
<td>Modlog.exe</td>
<td>2.0.5029.0</td>
<td>09-Sep-2005</td>
<td>00:10</td>
<td>155,408</td>
</tr>
<tr>
<td>Modlog.exe</td>
<td>2.0.5029.0</td>
<td>09-Sep-2005</td>
<td>00:10</td>
<td>86,566</td>
</tr>
<tr>
<td>Modlog.exe</td>
<td>2.0.5029.0</td>
<td>09-Sep-2005</td>
<td>00:10</td>
<td>96,016</td>
</tr>
<tr>
<td>Modlog.exe</td>
<td>2.0.5029.0</td>
<td>09-Sep-2005</td>
<td>00:10</td>
<td>356,692</td>
</tr>
<tr>
<td>Modlog.exe</td>
<td>2.0.5029.0</td>
<td>09-Sep-2005</td>
<td>00:10</td>
<td>256,692</td>
</tr>
<tr>
<td>Modlog.exe</td>
<td>2.0.5029.0</td>
<td>09-Sep-2005</td>
<td>00:10</td>
<td>122,040</td>
</tr>
<tr>
<td>OlCc.dll</td>
<td>5.0.2195.7793</td>
<td>09-Sep-2005</td>
<td>00:10</td>
<td>787,712</td>
</tr>
<tr>
<td>OlCc.dll</td>
<td>5.0.2195.7793</td>
<td>08-Sep-2005</td>
<td>00:10</td>
<td>64,192</td>
</tr>
<tr>
<td>OlCc.dll</td>
<td>5.0.2195.7793</td>
<td>09-Sep-2005</td>
<td>00:10</td>
<td>59,624</td>
</tr>
<tr>
<td>OlCc.dll</td>
<td>5.0.2195.7793</td>
<td>08-Sep-2005</td>
<td>00:10</td>
<td>840,808</td>
</tr>
<tr>
<td>OlCc.dll</td>
<td>5.0.2195.7793</td>
<td>09-Sep-2005</td>
<td>00:10</td>
<td>232,240</td>
</tr>
<tr>
<td>OlCc.dll</td>
<td>5.0.2195.7793</td>
<td>09-Sep-2005</td>
<td>00:10</td>
<td>8,108,124</td>
</tr>
<tr>
<td>OlCc.dll</td>
<td>5.0.2195.7793</td>
<td>09-Sep-2005</td>
<td>00:10</td>
<td>71,040</td>
</tr>
<tr>
<td>OlCc.dll</td>
<td>5.0.2195.7793</td>
<td>09-Sep-2005</td>
<td>00:10</td>
<td>398,608</td>
</tr>
<tr>
<td>OlCc.dll</td>
<td>5.0.2195.7793</td>
<td>09-Sep-2005</td>
<td>00:10</td>
<td>19,210</td>
</tr>
</tbody>
</table>

**REMOVED IN THE LATEST BUILD**

- _NDRSERVERCALL2
- _ELFDEEREVICEEVENTSOURCE
- _IEFY_HANDLE_RUNDOWN
- _ELFRCLOSEL
- _PTXCONTEXTHANDLESFORRECORD

**END OF OLD REMOVED**

**ADDED IN THE LATEST BUILD**

- _IMP_I_RPCBINDINGISC/client/local
- _STRINGCOPYWORKED
- _FIXCONTEXTHANDLESFORRECORD
- _PFN_I_RPCSESSIONONLYCONTEXTHANDLE

**END OF ADDED**
Vulnerabilities Found

MS05-051 – MSDTC Vulnerability – CAN-2005-2119
- After being disappointed with Windows 2000 Update Release 1
- MS05-051 fixed only 1 MSDTC vulnerability and a few others
- But what is "wcscpy(arg_28, pwszNULL_GUID)"

"A remote code execution and local elevation of privilege vulnerability exists in the Microsoft Distributed Transaction Coordinator that could allow an attacker who successfully exploited this vulnerability to take complete control of the affected system."
MSDTC (MS05-051)

• Heap Overflow: CRpcIoManagerServer::BuildContext

• Lack of input validation allows for overwrite of the ‘pszGuidOut’ argument with a null GUID string

• Attacks XP/2000 (BuildContextW opnum 7) as well as NT40 (BuildContext opnum 1).

• Interesting new string added: 'At least one of the buffers passed into BuildContext has an incorrect length.' (0x6DFDE24B)

• 4 new string length checks added

Found by Derek Soeder in a standard patch diffing session
Vulnerabilities Found

MS06-003 – Microsoft Exchange TNEF Issue
- After being disappointed with Windows 2000 Update Release 1
- Found multiple vulnerable functions
- But only 1 was reported in the advisory

"TNEF Decoding Vulnerability - CVE-2006-0002
A remote code execution vulnerability exists in Microsoft Outlook and Microsoft Exchange Server because of the way that it decodes the Transport Neutral Encapsulation Format (TNEF) MIME attachment. An attacker could exploit the vulnerability by constructing a specially crafted TNEF message that could potentially allow remote code execution when a user opens or previews a malicious e-mail message or when the Microsoft Exchange Server Information Store processes the specially crafted message. An attacker who successfully exploited this vulnerability could take complete control of an affected system."
TNEF (MS06-003)

• Vulnerability reported from MS Security team only mentions HrDecodeEncapsulation.
• Many other changes were released in the patch within different functions.
• Example: HrDecodeRecipTable
• new > 10000 (2710h) check after _WSTRM_Read call
• Potentially exploitable (demo)
• Also added (encoding) updates to not allow malformed outbound TNEF.

NOTE: This is one of MANY size/length checks added in MS06-003.
Demonstration

Demo
Demonstration

Demo time......
In Closing

• What you don’t know can hurt you.
• Relying only on signatures can hurt you.
• Full-Disclosure from vendors would help.
• This is not just a Microsoft issue.
  - Oracle
  - Apple
  - HP
  - IBM
  - Other (Linux?)
References

- OpenRCE.org – Reverse Engineering Community
- Sabre-Security – Professional binary tools
- IDAPython – Python interface to IDA plugin API
- IDA Palace – Random IDA goodness
- eEye Blink – Generic Endpoint Security

• Thanks: Derek Yoda Soeder, Barnaby “The Claw” Jack, Hugo The Puto
Questions