Security Engineering in Windows Vista

HITBSecConf2006 - Malaysia

September 18th - 21st 2006 :: Kuala Lumpur, Malaysia

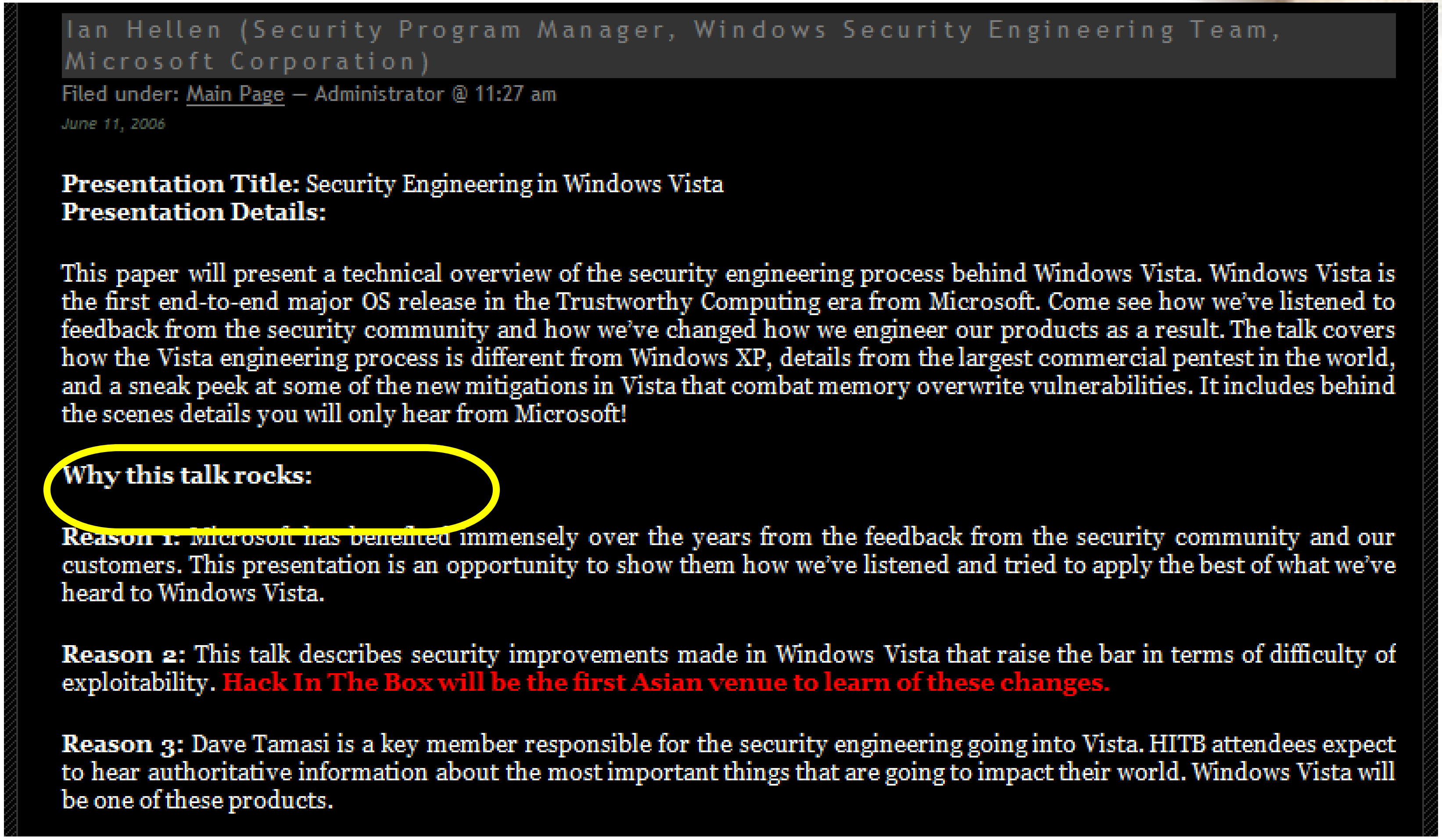
DEEP KNOWLEDGE SECURITY CONFERENCE

Ian Hellen & Vishal Kumar

Security Program Managers, SWI

Contact: ianhelle@microsoft.com vishalku@microsoft.com

A Public Retraction…



Ian Hellen (Security Program Manager, Windows Security Engineering Team, Microsoft Corporation)

Filed under: Main Page — Administrator @ 11:27 am

June 11, 2006

Presentation Title: Security Engineering in Windows Vista
Presentation Details:

This paper will present a technical overview of the security engineering process behind Windows Vista. Windows Vista is the first end-to-end major OS release in the Trustworthy Computing era from Microsoft. Come see how we've listened to feedback from the security community and how we've changed how we engineer our products as a result. The talk covers how the Vista engineering process is different from Windows XP, details from the largest commercial pentest in the world, and a sneak peek at some of the new mitigations in Vista that combat memory overwrite vulnerabilities. It includes behind the scenes details you will only hear from Microsoft!

Why this talk rocks:

Reason 1: Microsoft has benefited immensely over the years from the feedback from the security community and our customers. This presentation is an opportunity to show them how we've listened and tried to apply the best of what we've heard to Windows Vista.

Reason 2: This talk describes security improvements made in Windows Vista that raise the bar in terms of difficulty of exploitability. **Hack In The Box will be the first Asian venue to learn of these changes.**

Reason 3: Dave Tamasi is a key member responsible for the security engineering going into Vista. HITB attendees expect to hear authoritative information about the most important things that are going to impact their world. Windows Vista will be one of these products.

Intro – Who Am I (are we)?

Ian Hellen

- Security PM in SWI (Secure Windows Initiative)
- SWI is part of SECD (Security Engineering & Comms Dept.)
- Part of team of security reviewers working on Microsoft product security
- Ran Vista security design reviews
- Joined Microsoft 8 years ago
- 7 years in MCS – security in enterprise customers

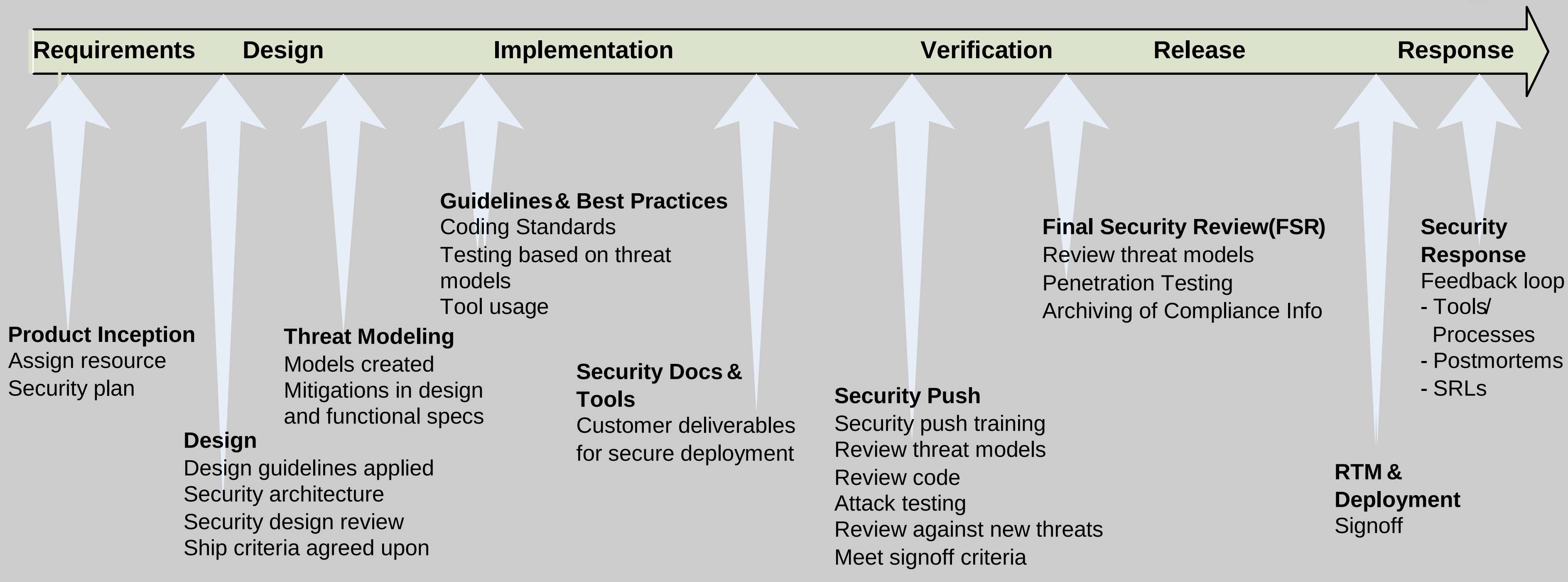
Vishal Kumar

- Security PM in SWI
- Responsible for Server and tools products for the past 2 years
- Now part of the design review team for the next Server Release
- Joined Microsoft 4 years ago
- Worked in the Core OS team prior to joining SWI

Goals of Talk

- Here to explain what we're doing in Windows Vista:
  - o Overview of security engineering activities
  - o Some detail on our major security initiatives
  - o Overview of our work on mitigations
- Here to listen to:
  - o Any engineering focused feedback you have
  - o How you think we're doing
- What you WON'T hear in this presentation:
  - o Security features

Security Deployment Lifecycle - Tasks and Processes



Product Inception
Assign resource
Security plan

Design
Design guidelines applied
Security architecture
Security design review
Ship criteria agreed upon

Threat Modeling
Models created
Mitigations in design and functional specs

Guidelines & Best Practices
Coding Standards
Testing based on threat models
Tool usage

Security Docs & Tools
Customer deliverables for secure deployment

Security Push
Security push training
Review threat models
Review code
Attack testing
Review against new threats
Meet signoff criteria

Final Security Review(FSR)
Review threat models
Penetration Testing
Archiving of Compliance Info

RTM & Deployment
Signoff

Security Response
Feedback loop
- Tools/ Processes
- Postmortems
- SRLs

Windows Vista Security Approach

Stop playing catch up! - Find & fix before ship!

Our Engineering Process:

1	**Apply least privilege throughout architecture**	Harden services, applications, browser
2	**Automate proven techniques**	Buffer overruns and common coding mistakes RPC and File parser fuzzing Banned API removal
3	**Methodically apply security expertise on whole product**	Attack Surface Reduction Threat Model reviews Design reviews Penetration testing
4	**Defense-in-Depth Mitigations**	Firewall on by default Enhanced protections for stack, heap, and more

Windows XP

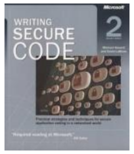
Security Bug Tracking

PREfix, Default Permissions

Across All Code

Component Team

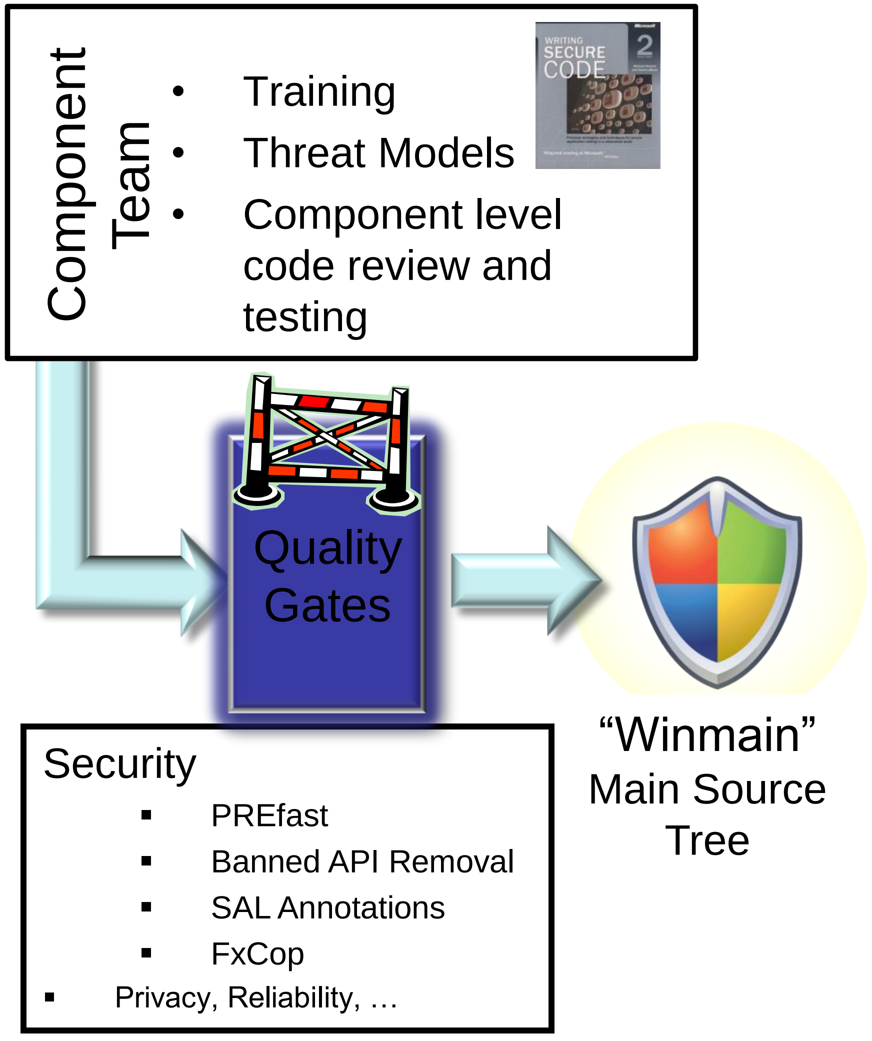
- Training
- Threat Models
- Component level code review and testing


WRITING SECURE CODE 2

Buddy Code Review

"Winmain" Main Source Tree

Windows Vista



Component Team

- Training
- Threat Models
- Component level code review and testing

Security

- PREfast
- Banned API Removal
- SAL Annotations
- FxCop
- Privacy, Reliability, …

“Winmain” Main Source Tree

Across All Code

- Security Bug Tracking
- PREfix, Default Permissions
- Threat Models Reviewed
- Legacy Banned API Removal & SAL Annotations
- Weak Crypto Removal, Service Hardening
- 3rd Party Code SDL Tracking
- Mitigations (/GS, etc)
- Privacy Review

High Risk Code

- Design & Attack Surface Review
- In Depth Threat Model Review
- Penetration Testing
- Mini-Security Push (if necessary)
- Network and File Parser testing
- Special Cleanup Projects

PART 1: APPLY LEAST PRIVILEGE THROUGHOUT ARCHITECTURE

Pervasive Least Privilege

- Problem: If a program runs as SYSTEM or Administrator, any compromise is catastrophic
- Approach:
  - Run Applications with Least Privilege
    - Enhance the standard user account
    - Administrators use full privilege only for administrative tasks or applications
    - Run some applications (browser) with heightened restrictions
      - See the IE7 Blackhat talk by Rob Franco for more information
  - Harden Services
    - Minimize privilege user pervasively in services
    - Define restrictions to ensure behavior conforms to expected activity

A Look at Service Hardening

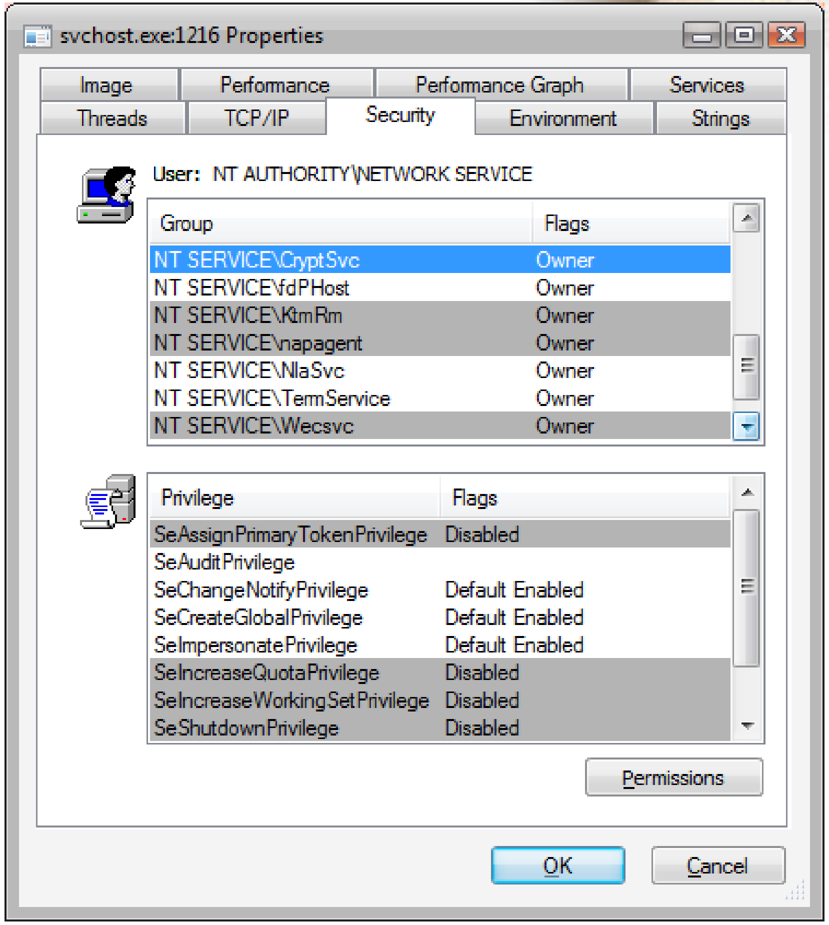
- Motivation:
  - o Services are attractive targets for malware
    - Sasser, Blaster, Slammer, Zotob, CodeRed
  - o No need for user interaction
  - o Often run in elevated identities
  - o Many worms:
    - Alter the OS
    - Open network connections to propagate

Reduce Privilege Level of Services

- Get services out of SYSTEM to LOCAL SERVICE or NETWORK SERVICE
  - LS & NS are not members of the Administrators group and aren't assigned the most powerful privileges like SeDebug, SeTcb, etc
- Reductions from Windows XP SP2
  - Eight (8) SYSTEM services now run as LOCAL SERVICE
    - Windows Audio
    - DHCP Client
    - Windows Event Log
    - COM+ Event System
    - Workstation Service
    - Windows Time
    - Security Center
    - Windows Image Acquisition
  - Four (4) SYSTEM services now run as NETWORK SERVICE
    - Cryptographic Services
    - Policy Agent
    - Telephony
    - Terminal Services
- **<u>And 48% of new services in Windows Vista run under a low privilege account</u>**

Compartmentalize Resources with Service SIDs

- Per Service SIDs
  - o Derived from service name in SCM
  - o S-1-5-80-xxxx
- ACL objects such that only your service can manipulate them
- Integrated into:
  - o LookupAccountSid
  - o LookupAccountName



User: NT AUTHORITY\NETWORK SERVICE

Group	Flags
NT SERVICE\CryptSvc	Owner
NT SERVICE\fdPHost	Owner
NT SERVICE\KtmRm	Owner
NT SERVICE\napagent	Owner
NT SERVICE\NlaSvc	Owner
NT SERVICE\TermService	Owner
NT SERVICE\Wecsvc	Owner

Privilege	Flags
SeAssignPrimaryTokenPrivilege	Disabled
SeAuditPrivilege	
SeChangeNotifyPrivilege	Default Enabled
SeCreateGlobalPrivilege	Default Enabled
SeImpersonatePrivilege	Default Enabled
SeIncreaseQuotaPrivilege	Disabled
SeIncreaseWorkingSetPrivilege	Disabled
SeShutdownPrivilege	Disabled

Example:
S-1-5-80-242729624-280608522-2219052887-3187409060-2225943459
Resolves to NT SERVICE\CryptSvc

Eliminate Unnecessary Privileges

- Describe the privileges your service needs and all others are removed
- Process that host multiple services get union of required privileges

```
// Set up the required privileges
SERVICE_REQUIRED_PRIVILEGES_INFOW servicePrivileges;
servicePrivileges.pmszRequiredPrivileges =
       (L"SeChangeNotifyPrivilege\0"
        L"SeCreateGlobalPrivilege\0"
        L"SeImpersonatePrivilege\0");

fRet = ChangeServiceConfig2(
        schService,
        SERVICE_CONFIG_REQUIRED_PRIVILEGES_INFO,
        &servicePrivileges);
```

Restricted Network Behavior

- Define the network requirements for your service…
  - You describe and OS enforces network access policy
  - Eg: foo.exe can only open port TCP/123 inbound
    - |Action=Allow|Dir=In|LPORT=123|Protocol=17|App=%SystemRoot%\foo.exe
  - If foo.exe has a bug, the rogue code cannot make outbound connections
- …enforced by the firewall

Group services to take advantage of restrictions

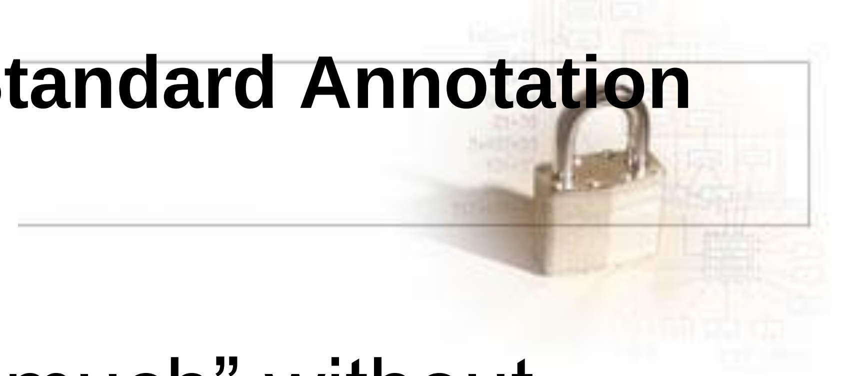
- New svchosts
  - **LocalServiceNoNetwork**
    - Runs under local service account. This group has no network access and has a write-restricted token
  - **LocalServiceRestricted**
    - Runs under local service account. This group uses a fixed set of network ports on the system and a write-restricted token
  - **LocalServiceNetworkRestricted**
    - Runs under local service account. This group uses a fixed set of network ports on the system but not a write-restricted token
  - **NetworkServiceRestricted**
    - Runs under network service account. This group uses a fixed set of network ports on the system and a write-restricted token
  - **NetworkServiceNetworkRestricted**
    - Runs under network service account. This group uses a fixed set of network ports on the system but not a write-restricted token
  - **LocalSystemNetworkRestricted**
    - Runs under local system account and accesses fixed set of network ports.

Case Study: DHCP Client Service

	Windows XP SP2	Windows Vista
Account	SYSTEM	LOCAL SERVICE
Privileges	24	4
Network Identity?	Yes (Machine Account)	No
Uses Fixed Set of Ports?	No	Yes
Data accessible only by service? (Service SID)	No	Yes

PART 2: AUTOMATE PROVEN TECHNIQUES

A Brief Introduction to the Standard Annotation Language (SAL)



- Tools can only find “so much” without more contextual information. Example:
  - o Given the code **`buff[x] = 5;`**
  - o How big is buff, and what is the value of x?
  - o C/C++ doesn’t associate buffers to their sizes
- SAL helps bridge the gap by providing interface contract information to the tools
  - o The concept is not new: think IDL in RPC
  - o Prime focus is finding buffer overrun bugs

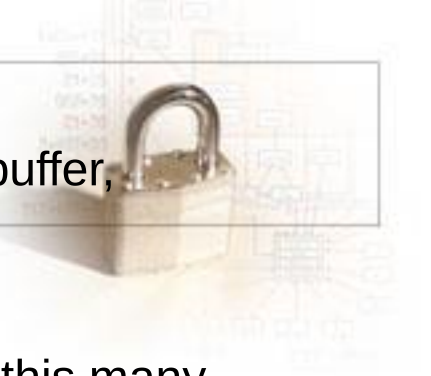
SAL Example

Pointer to a TCHAR buffer, we should annotate

Function writes this many characters to buf

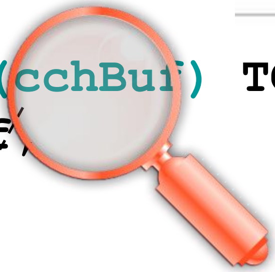
```
void FillString(
        TCHAR* buf,
        size_t cchBuf,
        char ch)
{
  for (size_t i = 0; i < cchBuf; i++)   {
    buf[i] = ch;
  }
}
```

- If cchBuf doesn’t correspond to the actual size of buf, the loop will walk off the end of buf



Annotations – A Closer Look

```
void FillString(
        __out_ecount(cchBuf) TCHAR* buf,
        size_t cchBuf,
        char ch)
```



```
_out_ecount(cchBuf)
```

Out parameter (will be written to) and is non-null

Buffer size is an Element count

Buffer is cchBuf elements in size

Adding Annotations

```
void FillString(
        __out_ecount(cchBuf) TCHAR* buf,
        size_t cchBuf,
        char ch)
{

  for (size_t i = 0; i < cchBuf; i++)    {
    buf[i] = ch;
  }
}

void main() {
        TCHAR *buff = malloc(200 * sizeof(TCHAR));
        FillString(buff,210,'x');
}
```

1. Warning C6386: Buffer overrun: accessing 'argument 1',
2. The writable size is ‘200*2' bytes, but '420' bytes might be written
3. The specification for the function. warning C6387: 'argument 1' might be '0': this does not adhere to 'FillString’ `__out`

Remember this Buffer Overrun?

- Buffer overrun found in IE7 Beta 2 on Jan 31, 2006.
  - o http://www.security-protocols.com/advisory/sp-x23-advisory.txt
  - o <BGSOUND SRC=file://--- >
- Workaround: Mozilla Firefox ☹

```
WCHAR pwzTempPath[MAX_PATH];

PathCreateFromUrlW(
   pwzPath,
   (LPWSTR) pwzTempPath,
   &cchPath,
   0);
```

PREfast & SAL in Action

```
LWSTDAPI
PathCreateFromUrlW(
        LPCWSTR pszIn,
        __out_ecount(*pcchOut) LPWSTR pszOut,
        __inout LPDWORD pcchOut,
        DWORD dwFlags )
```

```
WCHAR pwzTempPath[MAX_PATH];

PathCreateFromUrlW(
        pwzPath,
        (LPWSTR) pwzTempPath,
        &cchPath ,
        0);
```

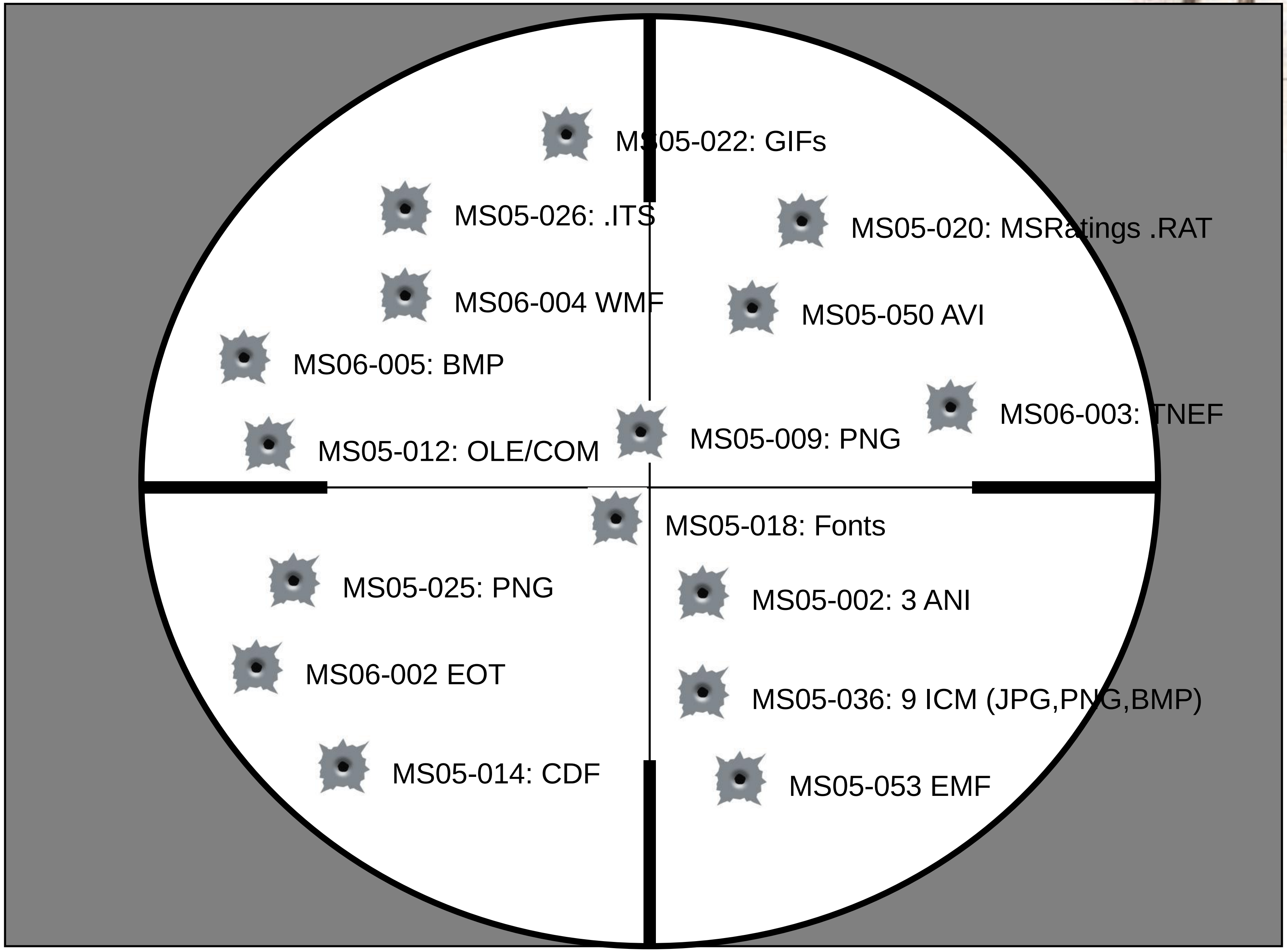
11/24/2005 5:50 AM Bug # ______ Opened by PREfast

1. Description: Potential overflow using expression '& pwzTempPath'
2. Buffer access is apparently unbounded by the buffer size.
3. In particular: cchPath`3485a is not constrained by any constant
4. Buffer is of length 260 elements (2 bytes/element) [size of variable or field]
5. Annotation on function PathCreateFromUrlW@16 requires that {parameter 2} is of length >= *{parameter 3} elements (2 bytes/element)
6. where {parameter 2} is & pwzTempPath; {parameter 3} is & cchPath

Did SDL succeed or fail?

- Root cause analysis leads to tools Improvement
  - After PnP RPC bug (Zotob worm), PREfast was improved (warning #2015)
- Process improved to auto-file bugs on #2015
- IE bug identified immediately and filed by PREfast toolset in Nov 2005
  - Caught by PREfast due to SAL annotation on PathCreateFromUrlW API
- Found through internal fuzzing efforts 8 days after public vuln report
- Reported through Windows Error Reporting 1 day later
- Bug was found and would have been fixed by RTM
  - Focus on root cause analysis, continuous tools and process improvements in SDL pays off

File Parsers: Under Attack

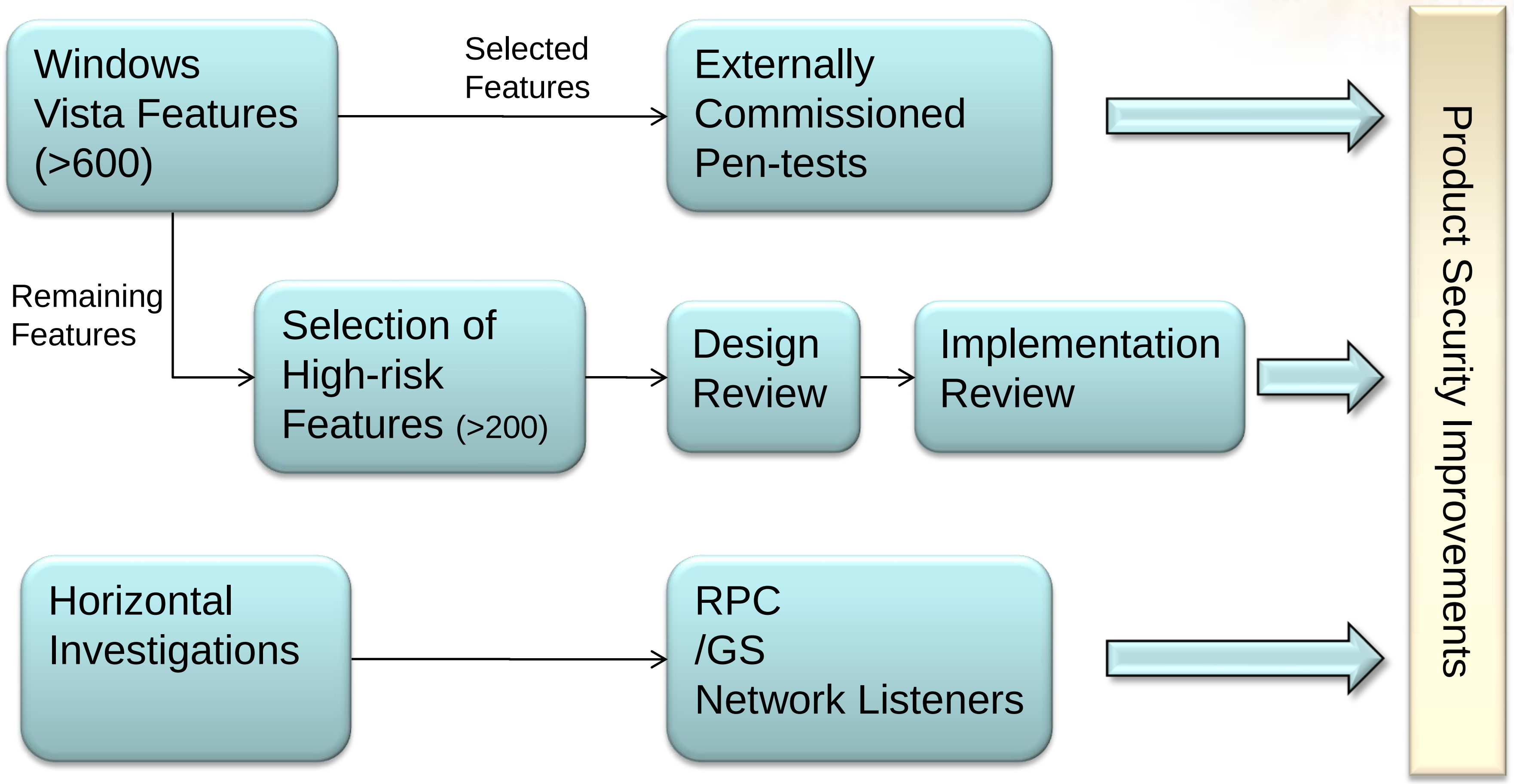


Multi-Prong Approach on Parsers

- Automate what you can:
  - o <u>All parsers</u>: Internally developed general purpose fuzzer
    - Over 100 Million manipulations by Beta 2
  - o <u>Highest risk parsers</u>: get Data-Definition-Language extensions
  - o <u>Hard targets</u>: Smart fuzzers (Examples: EMF, HTML)
  - o Code coverage helps in “template reduction” to improve efficiency
    - Library of >19,000 JPGs were optimized down to 47 with the same block coverage
- Apply security expertise where you need it:
  - o Security code review + detailed program analysis on “problem parsers”
  - o Extended SAL annotations for struct members
  - o Emit runtime stack protections more aggressively in “attack path”

PART 3: METHODICALLY APPLY SECURITY EXPERTISE

The Human Touch


Windows Vista Features (>600)
Selected Features
Externally Commissioned Pen-tests
Remaining Features
Selection of High-risk Features (>200)
Design Review
Implementation Review
Horizontal Investigations
RPC /GS Network Listeners
Product Security Improvements

Penetration Testing

- Largest Pentest in MS History
  - o Internal team of hackers (remember LSD?)
  - o Multiple simultaneous pentests (e.g. TCP/IP)
  - o “Blue Hat Hackers” :^)
    - 20+ security consultants (aka hackers) in a room
    - Access to Full Source + Symbols, specs, threat models
    - Access to members of product teams, SWI experts
    - All necessary expertise is within 1 building radius

  - o Spend anywhere from 1 week to 2 months per target
  - o Nothing is out-of-scope

Sampling of Findings

- Process tended to yield “rabbit holes”
- Contradiction in Security Assumptions
  - TM #1: “We have no risk because we don’t parse anything, we just pass things down.”
  - TM #2: “We have no risk because our input is trusted; we just receive validated content.”
- Failures of Imagination
- Unwise filenames

`wls0wndh.dll`

Winlogon Session0 Viewer Window Hook DLL

It Came from the Codebase…

```
n -= (e = (e = 0x4000 -
((d &= 0x3fff) > w ? d : w))
> n ? n : e);
```

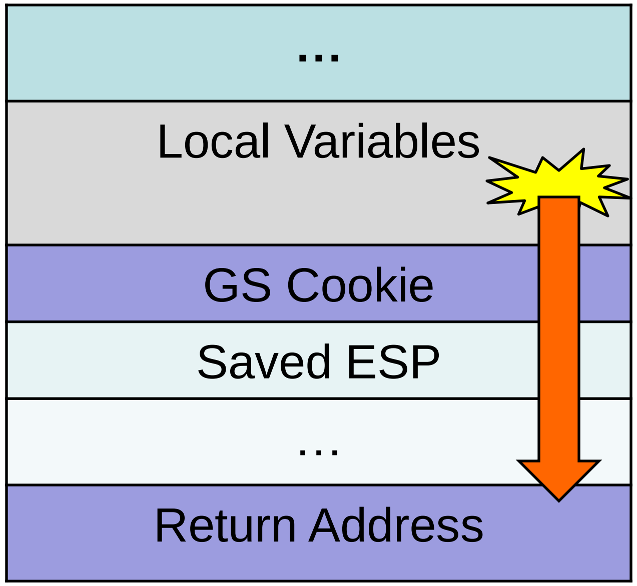
“It’s actually quite beautiful!
Almost like a Haiku or something.”

Found by Felix Von Leitner (n.Runs)

PART 4: DEFENSE-IN-DEPTH MITIGATIONS

Mitigations - /GS Flag

- /GS improved in MSVC 8.0 (Visual C++ 2005 aka Whidbey)
  - o Significantly improved /GS stack protection from Whidbey compiler
  - o Annotations force more aggressive protection in forward facing areas



Heap Hardening

- Many Defense in Depth changes:
  - Lookasides no longer used
  - Early detection of errors due to block header integrity check
  - Dynamic adjustment of algorithms based upon the usage (target attacks)
  - Pseudo-random base address
  - Heap TerminateOnCorruption
    - On by default for 64bit
    - On for services & most apps on x86.
      - still fine tuning
  - See Adrian Marinescu’s talk (Blackhat US 2006)

Function Pointer Encoding

- Function pointer encoding
  - Overwriting a function pointer in a predictable location is a common technique to gain control of EIP
  - Encode function pointer with secret; Decode prior to dereference
  - Decreases reliability of exploit by increasing chances of process termination due to AV on EIP, NX exception, invalid instruction, etc.
- APIs

EncodePointer	- XOR with a per process cookie
EncodeSystemPointer	- XOR with a shared cookie in SharedUserData

- Examples:
  - PEB Lock/Unlock routines
  - Pointers internal to NT heap
  - Vectored exception handler pointers
  - ~300 other usages as of Beta 2, typically long lived function pointers

Data Execution Protection aka NX

- Most Windows Vista PCs will have hardware support for NX
- Default Mode: Opt-in
  - o EXEs linked with /NXCOMPAT:YES have NX turned on permanently
  - o All Windows services and most EXEs will be opted in
    - Still battling compatibility issues with IE & Explorer and oddities with ATL thunk emulation
- Customers can put Windows in Opt-out mode
  - o Everything has NX turned on
  - o Exception list is configurable in registry

But DEP isn't Good Enough by Itself

As can be seen, the technique described in this document outlines a feasible method that can be used **to circumvent the security enhancements provided by hardware-enforced DEP…**

First and foremost, the technique depends on knowing the location of three separate addresses…**The first dependency could be broken by instituting some form of Address Space Layout Randomization** that would thereby make the location of the dependent code blocks unknown to an attacker.

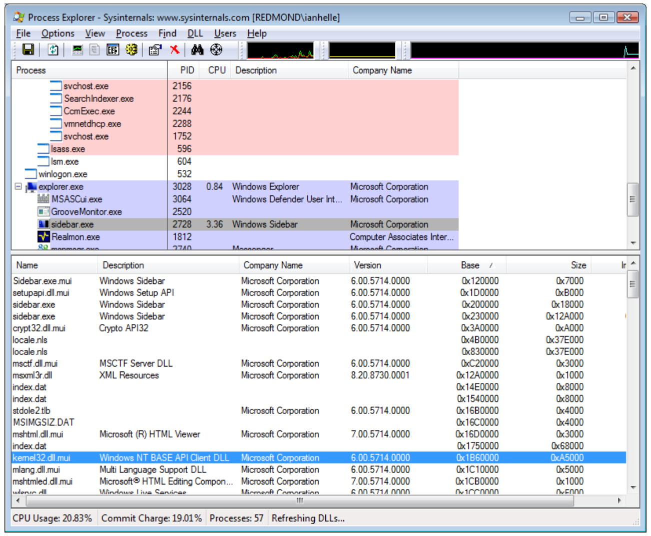
http://www.uninformed.org/?v=2&a=4
mmiller at hick.org, Skywing at valhallalegends.com

Mitigations – ASLR #1

- Address Space Layout Randomization (ASLR)
  - Powerful complement to NX (aka Data Execution Protection)
- Current (and probably RTM) design:
  - Images must opt-in via bit in PE header: DYNAMIC_BASE
    - New linker adds support; also emits reloc in EXEs
  - Limited # of bits available for randomness on 32bit.
    - Main trade-off
      - How difficult do you want the guess to be?
      vs.
      - How much contiguous virtual address space do you want to be available to apps?
    - Currently set so that 99.6% of the time, your first guess will fail
  - Still working through some issues: service restart, CLR

Mitigations – ASLR #2

- Impact:
  - No major hit on performance. Some wins. Some minor losses.
  - Application compatibility is looking good with current design (600 apps passed)
- On by default from Beta2
- All of Windows Vista is Opted-in ☺


Process Explorer - Sysinternals: www.sysinternals.com [REDMOND\ianhelle]
File
Options
View
Process
Find
DLL
Users
Help

Process	PID	CPU	Description	Company Name
svchost.exe	2156			
SearchIndexer.exe	2176			
CcmExec.exe	2244			
vmnetdhcp.exe	2288			
svchost.exe	1752			
lsass.exe	596			
lsm.exe	604			
winlogon.exe	532			
explorer.exe	3028	0.84	Windows Explorer	Microsoft Corporation
MSASCui.exe	3064		Windows Defender User Int...	Microsoft Corporation
GrooveMonitor.exe	2520			
sidebar.exe	2728	3.36	Windows Sidebar	Microsoft Corporation
Realmon.exe	1812			Computer Associates Inter...

Name	Description	Company Name	Version	Base	Size
Sidebar.exe.mui	Windows Sidebar	Microsoft Corporation	6.00.5714.0000	0x120000	0x7000
setupapi.dll.mui	Windows Setup API	Microsoft Corporation	6.00.5714.0000	0x1D0000	0xB000
sidebar.exe	Windows Sidebar	Microsoft Corporation	6.00.5714.0000	0x200000	0x18000
sidebar.exe	Windows Sidebar	Microsoft Corporation	6.00.5714.0000	0x230000	0x12A000
crypt32.dll.mui	Crypto API32	Microsoft Corporation	6.00.5714.0000	0x3A0000	0xA000
locale.nls				0x4B0000	0x37E000
locale.nls				0x830000	0x37E000
msctf.dll.mui	MSCTF Server DLL	Microsoft Corporation	6.00.5714.0000	0xC20000	0x3000
msxml3r.dll	XML Resources	Microsoft Corporation	8.20.8730.0001	0x12A0000	0x1000
index.dat				0x14E0000	0x8000
index.dat				0x1540000	0x8000
stdole2.tlb		Microsoft Corporation	6.00.5714.0000	0x16B0000	0x4000
MSIMGSIZ.DAT				0x16C0000	0x4000
mshtml.dll.mui	Microsoft (R) HTML Viewer	Microsoft Corporation	7.00.5714.0000	0x16D0000	0x3000
index.dat				0x1750000	0x68000
kernel32.dll.mui	Windows NT BASE API Client DLL	Microsoft Corporation	6.00.5714.0000	0x1B60000	0xA5000
mlang.dll.mui	Multi Language Support DLL	Microsoft Corporation	6.00.5714.0000	0x1C10000	0x5000
mshtmled.dll.mui	Microsoft® HTML Editing Compon...	Microsoft Corporation	7.00.5714.0000	0x1CB0000	0x1000

CPU Usage: 20.83% | Commit Charge: 19.01% | Processes: 57 | Refreshing DLLs...

SDL Case Study: Zotob Worm

Windows 2000	Remote unauthenticated code execution possible (No SDL prior to ship)
Windows XP SP1	Attacker requires authentication to exploit (ACL restricted)
Windows Server 2003	No remote security threat (Security RPC Callback added)
Windows XP SP2	No remote security threat (Reviewed and implemented Windows Server 2003 changes)
Even if we had missed it in Windows XP SP2	Blocked by firewall that is on by default

Zotob Worm in Vista?

Windows Vista	Improved PREfast & PREfix code scanners
And if that failed…	RPC Fuzzing and Penetration Testing
And if that failed…	Protected by an improved version of /GS & SafeSEH
And if that failed…	Protected by NX and ASLR
And if that failed…	And still blocked by firewall that is on by default

Don’t Miss...

- 13:15 Tomorrow
  Pen-Testing Windows Vista BitLocker Drive Encryption from the Inside
  Douglas MacIver

secure@microsoft.com

Questions?

HITBSecConf2006 - Malaysia

September 18th - 21st 2006 :: Kuala Lumpur, Malaysia

DEEP KNOWLEDGE SECURITY CONFERENCE