

# Exploiting Windows Vista: Protected Processes

Andy Ying, SIGMil

- ▶ Microsoft's newest operating system
- ▶ Build 6000 - known as Windows NT 6.0
- ▶ A bit bloated...
- ▶ Pretty user interface...
- ▶ Major kernel overhaul
- ▶ User Access Control and Patchguard
- ▶ DbgPrint are **NOT** sent to kd by default... (painful)
- ▶ *Protected Processes and Protected Media Path*

# What are Protected Processes?

- ▶ Made to provide the implementation of the protected media path.
- ▶ Created to satisfy the RIAA/MPAA's needs.
- ▶ You cannot inject a thread (no DLL injection).
- ▶ You cannot access the process's virtual memory.
- ▶ You cannot debug an active protected process.
- ▶ You cannot duplicate protected process' handles.
- ▶ You cannot impersonate a thread or set/retrieve context information.

## Previous Skirmishes

- ▶ The idea of attacking Vista's protected process concept is not new.
- ▶ Alex Ionescu published a tool called D-Pin Purr to do this in early April.
- ▶ However, D-Pin Purr was a binary blob with code obfuscation to prevent script kiddies from using this tool.
- ▶ Go to <http://www.alex-ionescu.com/?p=35> for more details...
- ▶ I did not reverse this blob because I didn't feel like reverse engineering some obfuscated code at 2:00am.
- ▶ Instead, I just hacked around the Vista kernel using WinDbg...

# Protected Media Path Technical Details

- ▶ Implemented in software.
- ▶ "The Protected Environment also provides all the necessary support for Microsoft-approved ("signed") third-party software modules to be added. It provides a wall against outside copying, where within the walls, content can be processed without making the content available to unapproved software." - Wikipedia
- ▶ "Windows Vista provides process isolation and continually monitors what kernel-mode software is loaded. If an unverified component is detected, then Vista will stop playing DRM content" - Wikipedia
- ▶ The protected process designator is one bit in a bit field in the process's EPROCESS structure.
- ▶ This is very, very weak in terms of implementation. It can be easily exploited...

```
kd> dt nt!_EPROCESS 835e2020
...
+0x224 RefTraceEnabled : 0y0
+0x224 NumaAware : 0y0
+0x224 ProtectedProcess : 0y1
+0x224 DefaultPagePriority : 0y101
+0x224 PrimaryTokenFrozen : 0y1
+0x224 ProcessVerifierTarget : 0y0
...
```

# DEMONSTRATION OF SIMPLE PROOF OF CONCEPT

# Malicious Applications

- ▶ Own1ng DRM applications.
- ▶ Using Vista to hide your spambot.
- ▶ Setting the ProtectedProcess bit in conjunction with unlinking its EPROCESS node to hide malware.
- ▶ Preventing anti-malware/anti-virus from analyzing your process's memory address space.
- ▶ Thwarting the World of Warcraft warden?

- ▶ Tested on Windows Vista Build 6000 x86 32bit mode on Virtual PC.
- ▶ Should work on any machine running the 32bit version of Vista.
- ▶ 64bit Vista will probably not work (at least not that easily).
- ▶ BUGS: Probably need to fix that APPCRASH in pmpctl.exe

Questions?