COSEINC®
Solid Security.Verified.

Phenom, *Bypassing Antiviruses*

COSEINC (AML) Advanced Malware Labs

Introduction

Modern Antiviruses rely on the modification of program code to redirect code execution flow. They achieve this by injecting interception code into programs before the original execution starts. There are different methods of redirection, such as employing user level or kernel level hooks. But for product stability, most of these Antiviruses still opt to rely on user level hooks. This is because kernel level hooks can crash the operating system when not implemented properly. In fact, some security software purely uses user level hooks to increase stability.

In recent years, we have encountered malwares which are aware of user level hooks and they utilized interesting techniques to bypass these hooks. More interestingly, some of these malwares also modified specific Antivirus engine. As a result, these Antivirus engine failed to read the malware memory address space.

Phenom

The Phenom malware is believed to be developed by skilled developers who are aware of Antivirus/HIPS protection techniques. It uses a known technique called '**Heaven Gate'** (also known as the '**0x33 segment selector'**) to bypass user mode hooks and sandboxes running under 32bit mode emulation on 64bit Windows. The malware variant that we were researching is the loader. This is the malware that performed the initial injection and bypassing. We did not research into any associated payload.

COSEINC®
Solid Security.Verified.

Heaven Gate

Windows 32bit compatible mode was designed to allow the execution of binaries compiled as 32bit under Windows 64bit. Processes which were compiled under 32bit are loaded inside the Windows-on-Windows (WoW64) subsystem and assigned running threads using 32bit compatible code.

Windows 64bit architecture calls, made under 32bit emulation mode, will eventually get translated and switch to full 64bit mode. This call translation is made through a specific segment call gate, commonly known as the 'Heaven Gate'. It enables every 32bit process to execute ***FAR CALL*** thorough this selector and switch to native 64bit mode.

Phenom uses this technique to bypass specific Antiviruses which uses 32bit specific hooking and Sandboxes. At the time of writing, most commercial sandboxes are prone to such bypassing.

```
************* Symbol Path validation summary **************
Response                         Time (ms)     Location
Deferred                                       SRV*c:\sym*http://msdl.microsoft.com/download/symbols
Symbol search path is: SRV*c:\sym*http://msdl.microsoft.com/download/symbols
Executable search path is:
ModLoad: 00000000`00400000 00000000`0047d000   image00000000`00400000
ModLoad: 00000000`77420000 00000000`775c9000   ntdll.dll
ModLoad: 00000000`77600000 00000000`77780000   ntdll32.dll
ModLoad: 00000000`73d90000 00000000`73dcf000   C:\Windows\SYSTEM32\wow64.dll
ModLoad: 00000000`73d30000 00000000`73d8c000   C:\Windows\SYSTEM32\wow64win.dll
ModLoad: 00000000`73d20000 00000000`73d28000   C:\Windows\SYSTEM32\wow64cpu.dll
(86c.b4c): Break instruction exception - code 80000003 (first chance)
ntdll!LdrpDoDebuggerBreak+0x30:
00000000`774ccb60 cc              int     3
0:000> bp main
Bp expression 'main' could not be resolved, adding deferred bp
*** Bp expression 'main' contains symbols not qualified with module name.
0:000> bp ntdll32!ZwTestAlert
*** Unable to resolve unqualified symbol in Bp expression 'main'.
0:000> lm
start             end                 module name
00000000`00400000 00000000`0047d000   image00000000_00400000   (deferred)
00000000`73d20000 00000000`73d28000   wow64cpu   (deferred)
00000000`73d30000 00000000`73d8c000   wow64win   (deferred)
00000000`73d90000 00000000`73dcf000   wow64      (deferred)
00000000`77420000 00000000`775c9000   ntdll      (pdb symbols)          c:\sym\ntdll.pdb\6192BFDB9F04442995FFCB0BE95172E12\ntdll.pdb
00000000`77600000 00000000`77780000   ntdll32    (pdb symbols)          c:\sym\wntdll.pdb\DCCFF2D483FA4DEE81DC04552C73BB5E2\wntdll.pdb
```

The picture above shows a 32bit application being debugged using WinDBG. We can see that two **ntdll** versions are present, one for the 32bit (ntdll32) version and one for the 64bit (ntdll).

For a process to jump from WoW64 Compatible mode into 64bit mode, the process will have to go through the 'Heaven Gate' located at code segment selector **0x33**, which also identifies the call gate inside the **GDT (Global Descriptor Table).**

This technique poses a challenge for those trying to debug Phenom using OllyDBG Debugger. OllyDBG is a 32bit application and when the researcher tries to debug Phenom, the debugger will lost its context when the malware issued a ***FAR CALL*** and jump directly to 64bit context.

This same issue will also affect Antiviruses which are only hooking under 32bit mode. The hooking will not intercept the ***FAR CALL*** and Phenom will continue to inject its payload without triggering the Antivirus engine.

Executing the Loader

The following steps describe how the loader operates.

1. *Align the stack on 64bit boundary.*

   Phenom is doing it since it needs to be aligned on 8 byte boundary from the beginning (Base Address). See example bellow.

Address	Element
0×00000000'00100000	StackBase+0 (0)
0×00000000'00100008	StackBase+8 (8)
0×00000000'00100010	StackBase+10 (16)
0×00000000'00100018	StackBase+18 (24)
…	StackBase+20 (32)
End of Stack	StackBase+X

2. *Phenom loads libraries which are located in the 64bit ntdll version*

   Phenom gets the libraries address by calling another function which receives as parameter the libraries name and returns the libraries Base Address Offset

   US_WindowsPS(

   LdrLoadDll,

   LdrGetKnownDllSectionHandle,

   NtFreeVirtualMemory,

   NtMapViewOfSection,

   NtUnmapViewOfSection)

3. *Phenom uses fail safe function*

The loader retrieved the function pointers from other libraries while using a different function named ***US_Windows64GetProcAddress.*** This function receives the module base address as first argument and the function API name as the second parameter. It then start to walk through the PE header of the first parameter's module and loads up the ***IMAGE_EXPORT_DIRECTORY*** to identify the function.

The export structure (***IMAGE_EXPORT_DIRECTORY)*** holds 11 members in its structure but not all the fields are utilized. Below are the structure members:

Field Name	Resolving
Name	Module name, mandatory because
Base	Number you must bias with the ordinals in order to get the indexes into the Address Space
NumberOfFunctions	Number of symbols which are exported by the Module
NumberOfNames	Number of symbols that are exported by name
AddressOfFunctions	RVA that points to an array of RVA symbols inside the module
AddressOfNames	RVA that points to an array of RVA of names and functions inside module
AddressOfNameOrdinals	RVA that points to a 16bit array that contains the ordinal associated with the functions

4. *Locate and load the 64bit version of kernel32.dll*

Phenom loader last phase is to locate and load the **64bit** version of **kernel32.dll.** The developer chooses a unique approach, which freed the memory location of **kernel32.dll** and **user32.dll** library base address and then loads each independently using the ***LdrLoadDLL*** function (with base address retrieved from ***US_WindowsPS*** function).

Payload

As mentioned above, we did not manage to retrieve any payload associated with Phenom loader. However, we did observe how the payload gets loaded:

Once the **kernel32.dll** gets loaded, the loader is ready to **load & execute** any external libraries which are being loaded using the **LoadLibrary** function. The library is being loaded using the ***CreateRemoteThread*** function API and this enables it to inject into ANY 64bit application.

Conclusions

Phenom is an interesting malware that uses 'Heaven Gate' and other interesting library loading techniques. This increases the complexity to fully reverse engineer and understand the malware. The Antivirus evading techniques that it uses is also innovative and we will probably see more malware utilizing this evasion method in the future.

The full functionality of the malware is still under research and we will provide full description soon.