Info Process Pty Ltd.
Sydney, Australia.

Windows Security and
Vulnerability Research

AntiHook:
Host Intrusion Prevention System by Info Process Pty Ltd.

Aug 29, 2006

Abstract
This paper provides information about Windows security vulnerabilities and the solutions provided by AntiHook – a Host Intrusion Prevention System from Info Process.

This information applies to the following operating systems:

Microsoft Windows Server™ 2003

Microsoft Windows XP

Microsoft Windows Vista

References and resources discussed here are listed at the end of this paper.

Contents

Disclaimer

This is a preliminary document and may be changed substantially prior to final commercial release of the software described herein.

The information contained in this document represents the current view of Info Process Pty Ltd on the issues discussed as of the date of publication. Because Info Process must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Info Process, and Info Process cannot guarantee the accuracy of any information presented after the date of publication.

This White Paper is for informational purposes only. INFO PROCESS MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

1 Introduction

In the last few years mass worm attacks, such as MS-Blaster, SQL Slammer and Code Red as well as Rootkits and Trojans have caused massive damage to individuals and organisations.

Today's threats and cyber criminals have advanced beyond the capabilities of any signature-based detection system. New vulnerabilities and new exploits provide attackers with a wide enough window of opportunity to launch malicious Rootkits and other sorts of malware that remain completely undetected by signature-based security software. It is not possible anymore to identify malicious threats and block the growing number of zero-day exploits and polymorphic malware by using standard Antivirus and signature based AntiSpyware applications alone.

Businesses are vulnerable until security vendors capture, analyse, and identify the new attack and deploy new signature definitions for their products.

Statistically it takes on average about six hours to find, classify, and push out a new definition to the user's desktop. The Achilles heel of the whole industry is that these research techniques can do nothing to protect the user against a custom virus or Trojan, not to mention malicious kernel level Rootkits. Custom malware is part of the new face of cyber threats, and hackers are targeting private user information and organisation's business information and assets. Custom malware also does not need to rely on vulnerabilities to execute and can appear as legitimate software. These threats can be completely undetectable unless a behaviour based HIPS is installed.

What makes this even worse is the fact custom malware is easy to create. There are plenty of code examples available on the net that show how to create a trojan, virus or even a low-level Rootkit, and modify it so that existing anti-virus and anti-spyware programs do not recognise it.

Info Process, through its security research is stepping up to the challenge with a range of technological approaches that closes this window of common vulnerability and provides a Managed Operating Kernel for the Microsoft Windows environment.

2 Why AntiHook?

AntiHook technology was developed after years of security research on Windows NT vulnerabilities and other type of attacks employing sophisticated user and kernel level hacking techniques for gaining control over a user's machine through compromising application isolation and loading malicious binaries.

The key benefits of the AntiHook architecture are:

- AntiHook is a **proactive kernel mode protection** that provides a Managed Operating System Kernel environment.
- AntiHook allows the system to be secured by **detecting and preventing attacks in real-time**; following the rule "prevention is better than cure".
- AntiHook is a behaviour based security framework. **No updates are required** and thus providing **zero-day protection** from new threats.
- AntiHook complements existing network firewalls, signature based virus protections and adds value to Windows built-in security features. In fact with the ability to prevent processes from being terminated or hooked it **provides protection to other security software** running on the Windows host.

- AntiHook is not just a security framework – with the ability to control the software that is authorised to run it also has wide ranging applications in the software control and licensing areas.

AntiHook is a highly functional behaviour based security framework designed with kernel mode protection that provides true process isolation and controls what software is running on a user's machine. In addition to this it also provides enhanced data protection, privacy and operating system integrity by monitoring the behaviour of all running processes and loaded drivers.

All attacks are detected and blocked in real-time. AntiHook detects and stops suspicious software launches or behaviours before they are allowed to occur.

The AntiHook solution is also unique in that it does not rely on lists of known malware, so no updates are required and no connection to a server or the internet is required either - the protection is virtually built into the operating system.

AntiHook is the pioneer in the new generation of products where the main goal is to effectively differentiate bad behaviour from good and not allow the compromising of the host or network from within by providing true application isolation. AntiHook technology is aimed at all Windows machines, including desktops, notebooks and servers, and provides a great experience to users and administrators alike.

AntiHook provides an extra layer of defence on critical servers and desktops, especially against new attacks that can bypass signature based security protection. It also acts as shield against potential new exploits during the crucial period between the initial circulation of malware and when an updated signature file or patch is released and installed.

As a behaviour based filter the AntiHook kernel agent is installed at the lowest possible operating system level. Any system calls made by applications are verified against rules and policies, or lists of known and configured behaviours. The kernel agent filters crucial system calls and can block any that are suspicious or that violate rules or policies. The outcome is that zero-day or unknown attacks can be blocked without having to wait for updated signatures to be developed and deployed.

AntiHook filters the majority of the operating system calls, but depending on the system call, the calling process, and the kind of protection policy applied, the kernel agent performs only the minimum necessary processing. This approach minimises the CPU overhead and improves the overall system peformance.

AntiHook is not just a security system – it is a framework that can be used for sandboxing environments, forensic examinations, enforcing workstation software and licensing control, and other administration functions.

AntiHook can be configured to isolate unknown or untrusted software before execution, thus enforcing corporate software policies, and preventing possible loss of information or damage to the user's machine from malware.

3 How Hackers Gain Control

The ultimate goal of the attacker is to obtain access to a host that is not normally granted, and to use this access to obtain sensitive information or files, or to exert a control over the host for malicious purposes. Attacks can range from large corporate espionage and theft, down to the stealing of information from a home user, possible leading to identity theft or obtaining internet banking access.

There are different types of Malware and they can be grouped in different categories based on their characteristics and the way they take control over a user's machine, but what they typically do is to alter operating system behaviour by adding or modifying user mode applications or the kernel of the system.

For example Trojans and Backdoors can add software to the system or in some cases mimic existing programs by physically replacing the binaries of the programs.

The Kernel of the operating system is where actually the software meets the hardware. Once the Kernel has been compromised, the underlying physical environment including all devices is under the control of the attacker. This subversive sort of malware are also known as kernel mode Rootkits. Typically these are small and very effective device drivers and because they are fully trusted by the operating system they can control every single aspect of the workstation or server. Rootkits work using a simple concept called modification of data or/and code execution which makes legitimate software make incorrect decisions.

More sophisticated kernel mode Rootkits use stealth techniques by design. They do not require a reboot to install, and they also survive reboots of the host. They can remain unseen to the user and other security products.

There are a few solutions available that help detect Rootkits, such as Rootkit Revealer and Black Light. These however have their limitations, both in their ability to detect Rootkits, but also in that they are detection devices, not preventions. There are also a few Rootkits on the net that make cheating products, like these a trivial task.

Malware can be grouped into three major groups:

- Malware which doesn't modify Windows in any undocumented way but manifests itself as additional software.
- Malware which alters the operating system behaviour by installing hooks or other types of remote code execution.
- Malware that modifies the data without actually changing the operating system code.

4 AntiHook Framework Features

AntiHook has been designed to ensure the integrity of the operating system and to protect against all types of Malware.

AntiHook technology can be configured to detect and allow, or respectively block, code execution in any of the three ways:

- A black-list of known code and behaviour that must be blocked is known, and everything else is allowed to run normally, or
- A white-list of known code and behaviours that must be allowed is known, and everything else is to be blocked, or
- A mixture of white-list and black-list can be defined, and any unknown code or behaviour is prompted for action.

AntiHook offers the following features:

Feature	Description
Trusted baseline with core Windows operating system.	AntiHook eliminates any race with Rootkits and Malware that have been installed before or after AntiHook as it takes a unique approach to retrieve and compute the core OS function entry points that might be compromised. AntiHook establishes a static trusted baseline with the operating system kernel. This technique ensures that all filters installed by AntiHook are called first before any other filters. AntiHook is therefore more advanced in this area compared to other anti-spyware products. AntiHook provides an effective defence that ensures that AntiHook will "get there first" even in the event a hidden Rootkit is already installed on the machine.
Self-protection and Self-recovery.	AntiHook can detect and block Rootkits and Malware which are attempting to disable AntiHook or to modify the Windows System Service Dispatch table or other Windows structures. It blocks malicious applications attempting to unload or stop the AntiHook driver and service…it protects itself! In addition to this AntiHook implements a sophisticated mechanism for recovering critical system structures that have been modified.
Starting of applications and programs.	Exact control can be had over what software is allowed to run. This feature allows policy implementation by placing restrictions on what applications execute on a user's machine.
Terminating and Suspending of critical security applications.	An important feature that keeps the integrity of the operating system by protecting critical security applications (e.g. Firewall, Anti-virus, and Anti-spyware software) from being terminated or disabled. AntiHook ensures integrity in all layers of security on the host by protecting the other security applications from attack.
Blocks hiding of critical OS components.	The key point of hiding a process is that it allows an attacker to create an undetectable backdoor that cannot be discovered using standard process tools like Process Explorer or Windows Task Manager or even Anti-rootkit software. For example the Hacker Defender Rootkit installs hidden backdoors, registers as a hidden system service and installs a hidden system driver. AntiHook can detect and block the hiding of operating system critical components such as processes, drivers and DLL's.
Stop attacks intended to modify processes.	Malicious software can execute code in a remote program and perform remote injection of code through standard Win32 API or other native APIs called from within Rootkits or other malicious programs. AntiHook provides protection against this. Here are some of the areas of the process modification monitoring functionality: • Opening a process handle with writable access to its code and data memory through Win32 APIs

Feature	Description
	OpenProcess and its kernel mode equivalent ZwOpenProcess • reading/writing to process memory through Win32 APIs ReadProcessMemory, WriteProcessMemory and their kernel equivalents • reading/modifying process module information • creating remote threads • opening and manipulating thread handles • attaching a debugger to processes • suspending the execution of a thread in an external process • setting the execution context of a thread in an external process
Ability to detect and block kernel and user land Rootkits.	All known Rootkits, and known Rootkit techniques, are detected and blocked. If the AntiHook driver is installed after a Rootkit it still allows the detection of the Rootkit.
Service/driver installation and loading of kernel drivers.	AntiHook detects the installation of kernel mode Rootkits that integrate and then maliciously alter the Windows operating system. It also detects and can stop loading of drivers which are being loaded through a call to ZwSetSystemInformation (SystemLoadAndCallImage) native API.
User mode hooking by modifying IAT or EAT.	AntiHook can detect and stop user mode IAT/EAT hooking by filtering open process requests any attempts to write to an external process memory
Start-up applications and DLL's	AntiHook can detect the registration of programs and DLLs for loading on PC start-up or when the user logs on to the system.
System-wide Windows Hooks	Malicious software can install system-wide Windows hooks by using standard Win32 APIs or native APIs. AntiHook can detect and prevent this by monitoring if software attempts to hook SetWindowsHookEx, NtUserSetWindowsHookEx or all related API's.
Registry modifications	AntiHook can protect a user's machine by detecting and stopping applications which are attempting to modify critical registry settings.
Attaching a debugger to another process.	AntiHook can detect and prevent a debugger being attached to a process. This solution can protect the user's machine from any sort of "debugging related" attacks.
Event Logging	Full logging functionality is provided, facilitating consistent logging and instrumentation practices, both within an application and across the enterprise. AntiHook can log to a local store, or integrate with the MS Enterprise Library's Logging Application Block as well as SQL databases.
Identifying the origin of suspicious high privileged threads running in the System Process	AntiHook can detect and suspend suspicious kernel drivers that have created system threads to compromise the operating system.

5 Extensibility

AntiHook is not only highly configurable, the AntiHook technology has also been designed with extensibility in mind. The architecture is a layered approach with the

ability to allow plug-in custom components for extending or replacing the functionality of the framework. There are a variety of mechanisms that can be utilised to integrate at different levels, thus allowing AntiHook to be incorporated into overall applications and solutions, or enterprise wide platforms.

6 Considerations

The cost of implementing behaviour based security software is that the host system must be well trained to reduce the number of false positives and generated “noise”.

AntiHook with its configuration and extensibility features can be implemented in a way to minimise the time required to setup and deploy, and once AntiHook has been well trained it becomes fully transparent to the user.

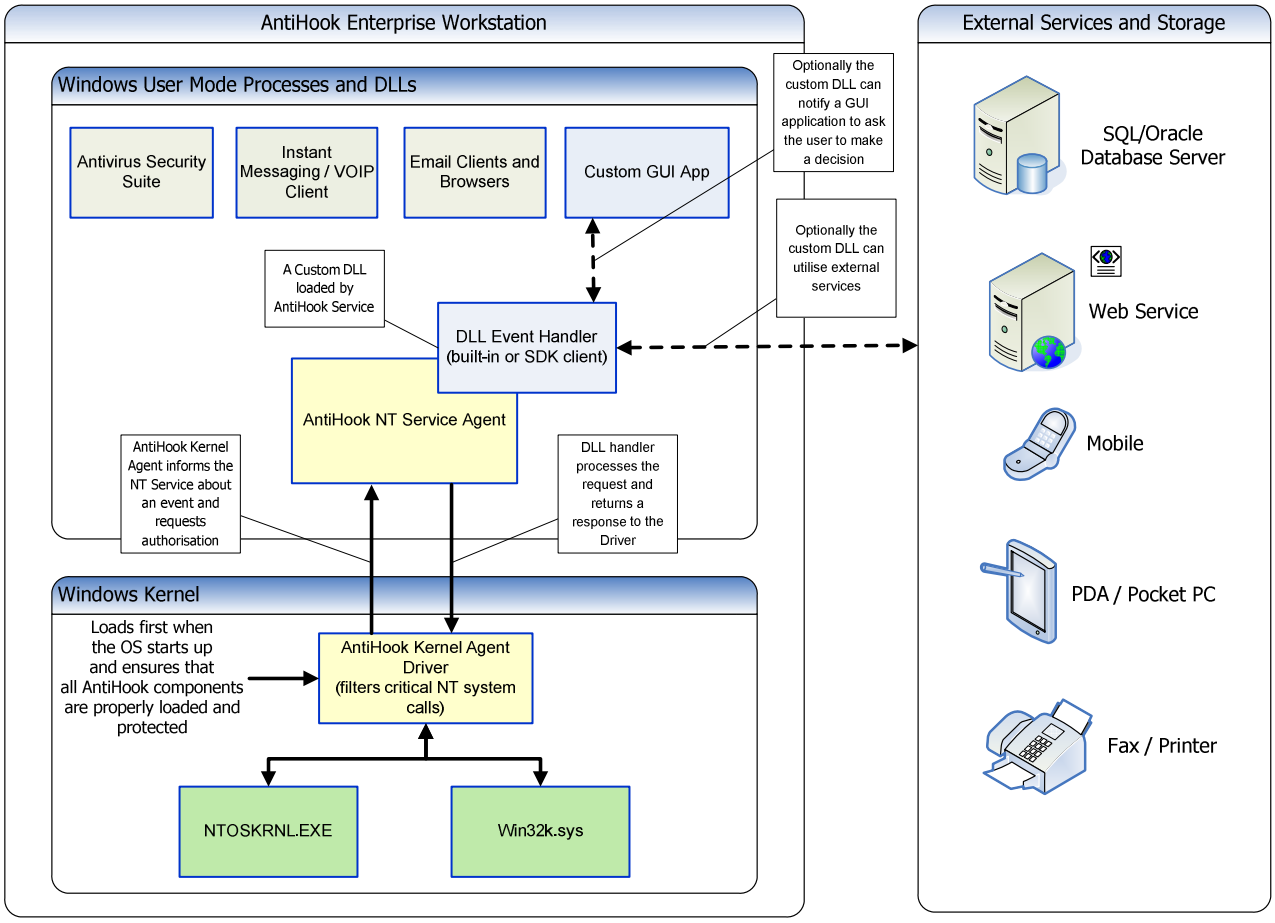
7 Architecture

AntiHook can be deployed on servers, desktops and portable computers. It is in fact a kernel agent written as a kernel-mode driver that installs itself at the lowest possible kernel level. It is responsible for intercepting requests for executables to launch, terminate or interact with other applications. The result is that any unauthorised program, including malware that tries to load itself onto the computer, is blocked.

The AntiHook kernel agent is typically installed on the destkop and can be controlled from a central management console where system administrators can create all the rules and configure policy. The AntiHook agents can also log every event that occurs on the desktop, thus giving administrators a global view of application activity within the organisation.

Unlike other security software AntiHook installs a filter that inspects crucial system calls and can block any that seems to be suspicious or that violate rules or policies.

The AntiHook core has been designed like an SDK and exposes an API interface to allow integration with other security products that can utilise Win32 API or .NET interfaces. This approach allows third party products to be easily plugged-in and to subscribe to notifications of specific or all native system calls which AntiHook filters.



AntiHook eliminates any race with Rootkits and Malware that have been installed before or after AntiHook as it takes a unique approach to retrieve and calculate the core OS function entry points that might be compromised.

AntiHook establishes a static trusted baseline with the operating system kernel. This technique ensures that all filters installed by AntiHook are called first before any other filters.

AntiHook is therefore more advanced in this area compared to other anti-spyware products. AntiHook provides an effective defence that ensures that AntiHook will “get there first” even in the event a hidden Rootkit is already installed on the machine.

For a complete HIP system it is imperative the design of the security solution is both efficient and provides an isolated and secure managed core environment. This is why one of the major goals of the design of AntiHook has been to secure the host and to maintain operating system integrity by providing a Managed Operating Kernel environment.

8 Threat Modelling

Threat modeling used in the design of the AntiHook architecture is based on years of security research and it is in fact a constant ongoing process. This has research has helped Info Process to systematically identify and rate the threats that are most likely to affect Windows kernel and user mode applications.

Info Process' understanding of Rootkit technology plays a key role in the design of the AntiHook Framework. By identifying and rating threats based on a solid understanding of the architecture and implementation of the operating system, threats are addressed with appropriate countermeasures by providing a Managed Operating Kernel.

To ensure that AntiHook does actually provide the best available protection of the host environment, the technology has been thoroughly tested against all Rootkits available on rootkit.com and phrack.org, as well as other advanced Leak Testers and malicious software examples.

9 Protecting other Products

AntiHook techology has been tested and proven to be complementary with the vast majority of the AntiVirus and Firewall products.

One of AntiHook's major goals is to protect other software from being compromised. It's been designed to detect and block any attempts to modify IAT and EAT. This protection technique is achieved by providing a stronger process isolation built inside the AntiHook kernel mode agent.

AntiHook is able to detect and stop any user mode intervention and stop any hooking of Win32 API calls in user-mode by ensuring process isolation and providing Managed Operating Kernel environment. This approach allows third party DLLs to run in a completey safe environment.

In addition to these features AntiHook filters all native operaring system calls that allow one process to read or write from/to the memory of an external process.

AntiHook is a highly configurable framework and can be setup to filter a group of, or all, native operating system calls.

10 AntiHook and Web Security

Web application security must be addressed across the tiers and at multiple layers. A weakness in any tier or layer makes a Web application or service vulnerable to attack.

It is imperative that security design covers the securing of the client, the network, the server host, and securing the applications, to provide sufficient levels of protection to ensure that no tier gets compromised.

For an example, even if a user has an encrypted SSL connection between their browser and a host server, this is no protection at all from any malware resident on their own computer, such as a key-logger recording the key strokes before the information is encypted and sent via the net.

AntiHook adds host based protection for the operating system and applications on both the clients and server hosts in a web based environment.

11 Microsoft Initiatives and Challenges

11.1 Microsoft Next-Generation Secure Computing Base (NGSCB)

Formerly known as Palladium, this is a software architecture designed by Microsoft which is supposed to address "Trustworthy Computing" concept on Windows 2003 64-bit and Vista 64-bit. NGSCB relies on hardware technology designed by members of the Trusted Computing Group, which provides a number of security-related features in conjunciton with a kernel level software implemented by Microsoft.

The operating system security kernel (Nexus) bridges the gap between the hardware security chip and the application security components. It checks that the hardware components are on the TCG approved list and the software components have been signed, and that none of them has a serial number that has been revoked.

NGSCB would make Windows more secure, but it can be still compromised since it relies on a kernel software component that is not fully protected. AntiHook adds value by providing additional lower-level kernel protection that guards and keeps the integrity of the operating system.

11.2 PatchGuard

The x64-based versions of Microsoft Windows 2003 , Windows XP, and Windows Vista for x64-based systems implement an additional layer of protection to stop the kernel from being patched except through authorised Microsoft-originated hot patches.

PatchGurard is intended to prevent both malicious software and third-party vendors from modifying certain critical operating system structures. These structures include things like specific system images, the SSDT, the IDT, the GDT, and certain critical processor MSRs. This feature is intended to ensure kernel stability by preventing unwanted behavior, such as hooking.

At a high-level, PatchGuard is implemented in the form of a set of routines that cache known-good copies and/or checksums of structures which are then validated

at certain random time intervals. There have already been a few papers published on how to bypass PatchGuard.

Whilst Microsoft have made a commendable effort with PatchGuard, there are still ways around the extra protection that are likely to be exploited by the more determined hackers. Due to the fact that PatchGuard in effect is a third-party driver running in the same protection domain as other drivers, including malware, then there is no guarantee that PatchGuard is not compromised. It is believed that Microsoft's approach of obfuscation will not provide a sufficient detterrant to hackers.

11.3 Blue Pill

At the Black Hat 2006 conference, Joanna Rutkowska, security researcher and a member of rootkit.com, showed that she found a way to bypass the Vista integrity-checking process for loading unsigned code into the Vista kernel. Then she presented **Blue Pill**, a rootkit she created based on Advanced Micro Devices (AMD) Secure Virtual Machine.

This demonstrates that even though PatchGuard and other built-in security techologies are very effective kernel guards, the operating system still requires an additional efficient kernel protection that is based on the technology which AntiHook implements to provide a Managed Operating Kernel.

12 Real Life Examples

HackerDefender

For the last two years a Rootkit called HackerDefender has gained a deserved reputation among the hackers for being one of the most effective Rootkits.

BackOrifice

A few years ago another Rootkit BackOrifice was quite popular as a customisable remote access tool that has legitimate purposes for security researchers, but also has been used by hackers.

Sony

Another controversial example is the alleged Sony CD copy protection DRM scandal dealing with Sony BMG Music Entertainment's stealthy distribution of software on audio CDs.

The software was proven to be a rootkit by design as it was hiding files, registry keys and processes with names starting with the string sys. The major problem with this was it made it very easy for writers of worms and other malware to utilise it and also hide their files by simply using the same naming convention. Within weeks there were several Trojans and worms taking advantage of this security hole in machines already compromised by the Sony rootkit.

The AntiHook Solution

In all of these examples the protection offered by AntiHook technology provides both detection and prevention of these attacks.

Systems protected with AntiHook were already protected from these malware examples before they were released (Zero Day protection) without the need for any signature update.

13 Resources

API Hooking Revealed by Ivo Ivanov
http://www.codeproject.com/system/hooksys.asp
How to build a user mode Win32 API spying system
Injecting techniques.
Interception mechanisms.

Rootkit.com
http://www.rootkit.com
One of the main Web resources for Black Hats, free Rootkit articles, code snippets, discussions and news.
Kernel and user mode Rootkits - Hacker Defender, HE4Hook, Vanquish. NT Rootkit, FU, WinlogonHijack, klister, Patchfinder2, VICE, NtIllusion, RAIDE etc.

Phrack
http://www.phrack.org
Digital hacking magazine.

Microsoft Next-Generation Secure Computing Base - Technical FAQ
http://www.microsoft.com/technet/archive/security/news/ngscb.mspx

Sony, Rootkits and Digital Rights Management Gone Too Far
http://www.sysinternals.com/blog/2005/10/sony-rootkits-and-digital-rights.html

Books

Exploiting Software
Greg Hogland and Gary McGraw. Addison Wesley, 2005.

Rootkits: Subverting the Windows Kernel
Greg Hoglund, Jamie Butler. Addison Wesley, 2004.

Undocumented Windows 2000 Secrets
Sven B. Schreiber. Addison Wesley, 2001.

Undocumented Windows NT
Prasad Dabak. M & T Books, 1999.