

Host Protection Technology Study

- Spyware Scanner** – Spyware scanners are designed to identify, remove, or block spyware. Where a virus scanner tends to look more for the initial infection payload, spyware scanners concentrate on recognizing the components of the attacks. Spyware scanners typically remove the threat by file deletion or registry key removal.

Conclusion

This study indicated defense-in-depth is a reliable strategy for defending a host. By overlaying the results of individual technologies in combination with each other, the potential that each combination had in defending against the various attack scenarios was examined. Indeed, many commercial host security suites take advantage of multiple technologies. One limiting factor

for most, if not all, organizations is the cost (monetary as well as time spent configuring and monitoring products).

Based on this study, Configuration 1 below is considered a good basic security configuration. As additional technologies are added to the system the amount of risk assumed is reduced, but cost increases. Ultimately each organization must make risk decisions as to the correct level of protection for their environment.

Finally, all organizations should utilize proven techniques to keep their networks and hosts secure. Applications and operating systems should be kept patched. Administrative accounts should only be used for activities that require administrative credentials. Users should be trained in proper security procedures for network access and should renew that training regularly.

Table 2: Protection Technology Configurations Comparing Risk and Cost

	Protection Technologies Applied	Risk	Cost
Configuration 1	Network Interface Firewall, Virus Scanner, Buffer Overflow Protection	HIGHER	LOWER
Configuration 2	Network Interface Firewall, Virus Scanner, Buffer Overflow Protection, Program Execution Blocking	↔	↔
Configuration 3	Network Interface Firewall, Virus Scanner, Buffer Overflow Protection, Program Execution Blocking, Process Monitoring	↔	↔
Configuration 4	Network Interface Firewall, Virus Scanner, Buffer Overflow Protection, Program Execution Blocking, Process Monitoring, File Integrity Monitoring, Registry Monitoring	LOWER	HIGHER

Introduction

Information is at the heart of most organizations' critical assets. Frequently, this information is either stored on, or accessed from, individual host computers connected to the organization's network. There are many technologies available designed to protect these hosts, but frequently network administrators have little insight about how the technologies perform against various attacks.

This study examined the behavior and capabilities of the different technologies against various attack scenarios in order to determine their effectiveness. The various products were grouped into different technology categories (host firewall, virus scanner, etc.) and installed in a controlled environment. A sample set of attack scenarios was constructed and tested against the various technologies. Because of the pervasive use of Microsoft Windows on desktop hosts, this study focused on Windows attacks and protection tools. Results were recorded and analyzed and a summary is presented here.

Protection Technology Definitions

- Buffer Overflow Protection** – Buffer overflows are common vulnerabilities that an attacker can exploit to disrupt service or to execute exploit code. Buffer overflow protection tools monitor a system's memory looking for misbehaviors such as a program executing from its stack space or a program executing from outside its declared memory space.
- Network Interface Firewall** – Network interface firewalls place a barrier between the host and external systems. All information traveling to and sometimes from the computer must pass through the host firewall prior to being fully processed. Ingress firewalls perform stateful
- Sandbox** – A sandbox attempts to create an isolated environment that controls the resources a program can manipulate. Its main function is to protect the operating system's critical resources, like the file system and the registry. The sandbox's isolated environment is achieved by virtualizing the applications' interaction with the OS. The sandbox technologies used for this study virtualized the applications, not the operating system.
- Program Execution Blocker** – A program execution blocker provides a layer of access control protection to the underlying Operating System (OS). As the name implies, its purpose is to allow or block execution of programs on the OS. Many program execution blockers can be



The Information Assurance Mission at NSA

configured to block execution based on user groups or particular user accounts for a more granular control. They utilize a list to make the decision of what to allow or block. There are two types of lists a program execution blocker can employ, a whitelist or a blacklist.

- Registry Monitor** – Most configuration information for a Windows system is found in the registry. Registry monitors exist to monitor and

protect this crucial part of the Windows operating system. Some monitors take a snap shot and then can be scheduled to run later to compare for unwanted changes. Other monitors try to protect the registry in real-time by intercepting any program trying to change the registry.

- File Integrity Monitor** – Much like a registry monitor, the file integrity monitor reports on changes to critical system and application files.

More sophisticated file integrity monitoring can also have the ability to prevent changes to critical system files without user interaction or to roll back the altered files to a safer state. In fact, most integrity monitors have the ability to protect both the registry and file system.

- Virus Scanner** – Virus scanners are designed to identify and handle malicious software. Virus

scanners examine files to locate known viruses, worms, phishing attacks, and other malicious code. Scanning can be performed anytime an application accesses a file or when scheduled by the user. Identification can be done through the use of a signature dictionary or through heuristic analysis. Identified viruses can be cleaned from the infected program; or the infected program can be deleted or quarantined.

Table 1: Protection Technology vs. Attack Vector Scenarios

Attack Vector	#Scenarios	Protection Technology									
		Buffer Overflow Protection	Network Interface Firewall	Network Interface IDS	Process/Application Behavior Monitor	Sandbox	Program Execution Blocker	File Integrity Monitor	Registry Monitor	Spyware Scanner	Virus Scanner
BO in IE--Connect back	5	Likely	Possible	Possible	Block	Possible					Possible
BO in IE--Connect back--Persist Reg	1	Block	Possible	Block	Block						Possible
BO in service--Command exec	1	Possible	Block	Partial	Block						
BO in service--Connect back	4	Block	Block	Possible	Block						
BO in service--Remote shell server	4	Block	Block	Possible	Likely						
BO in service--Remote shell server--Persist File/Reg/Service	1	Block	Block	Block	Block			Detect	Detect		
Race condition in IE--Connect back	1	Possible	Possible	Possible	Block	Detect					
Integer overflow in IE--Remote shell server	1	Block	Possible	Possible	Block						Block
Web dwndl--BO in Winzip--Command exec	1	Possible		Partial							
Web dwndl--BO in Winamp--Connect back	1	Possible	Possible		Possible						
Web dwndl--BO in Excel--Remote shell server	1	Block	Block	Possible	Block	Block					Possible
Code logic error in Firefox--Command exec	1	Possible		Possible	Block						Possible
Code logic error in IE--Persistent User startup--Connect back	1	Possible	Possible	Block	Block	Block	Detect			Possible	Possible
Code logic error in IE--Connect back--Remote shell server--Persist File/Reg/Service	1	Block	Block	Possible	Block	Block	Detect	Detect			
E-mail dwndl--User exec--Persist File/Reg/Service	1			Possible	X	Block	Detect	Detect			Block
E-mail dwndl--User exec--Persist File/Reg/Reg startup/WC--Adware	2			Possible	X	Block	Detect	Detect	Possible	Possible	
E-mail dwndl--User exec--Persist File/Reg--E-mail prop	1			Possible	X		Partial	Partial			Block
E-mail dwndl--User exec--Persist File/Reg/Reg startup/Service/WC--E-mail prop	1			Possible	X		Partial	Detect	Possible	Possible	
Web dwndl--User exec--Persist File/Reg/Service	2			Possible	Partial	Block	Block	Partial	Detect	Possible	Block
Web dwndl--User exec--Persist File/Reg/Reg startup	3			Possible	Possible	Block	Block	Detect	Partial		Possible
Web dwndl--User exec--Persist File/Reg/Reg startup--Adware	1			Partial	Possible	Block	Block	Detect	Partial	Block	Possible
Web dwndl--User exec--Persist File/Reg/Reg startup/WC--Adware	4			Possible	Block	Block	Detect	Detect	Likely	Possible	
Web dwndl--User exec--Persist Reg/WC--Adware	1			Possible	Block	Block				Possible	Block

Attack Abbreviations:
 BO = Buffer overflow
 dwndl = download
 IE = Microsoft Internet Explorer
 prop = propagation
 Reg = Windows Registry

Block	For attack vector, each scenario is blocked by all tools in technology category
Likely	For attack vector, most scenarios are blocked by all tools in technology category
Possible	For attack vector, at least one scenario is blocked by at least one tool in technology category
Detect	For attack vector, no scenarios are blocked but each scenario is detected by all tools in technology category
Partial	For attack vector, no scenarios are blocked but at least one scenario is detected by at least one tool in technology category
X	All scenarios in attack vector missed (neither blocked nor detected) by all tools in technology category
	All scenarios not tested

This study categorized host-based protection technologies into ten categories. In general, two tools were used to cover each category. Forty attack scenarios were constructed using publicly available exploits. When possible, multiple attack scenarios in each category were used. The table above details the summarized results from testing each tool against each attack vector scenario.