

SOPHOS FINAL INVALIDITY CONTENTIONS

EXHIBIT 505-A-22

Prior Art Reference: U.S. Patent No. 7,779,472 (“Hoglund”)

Hoglund was available at least as of June 22, 2005 and therefore qualifies as prior art under at least 35 U.S.C. §§ 102(a), (b), and/or 103 as to claims 1-20 (the “Asserted Claims”) of U.S. Pat. No. 8,856,505 (“the ’505 patent”).¹ Hoglund, including any material incorporated by reference into Hoglund, anticipates the Asserted Claims under 35 U.S.C. § 102 and/or renders obvious the Asserted Claims under 35 U.S.C. § 103, alone based on the state of the art and/or in combination with one or more references identified in Sophos’s invalidity contentions.²

To the extent Plaintiffs allege that Hoglund does not disclose any particular limitation of Asserted Claims, either expressly or inherently, it would have been obvious to a person of ordinary skill in the art as of the priority date of the ’505 patent to modify Hoglund and/or to combine the teachings of Hoglund with other prior art references, including but not limited to any systems relating to Hoglund or any other prior art

¹ Sophos may rely upon uncited portions of Hoglund, other publications, and the testimony of experts or fact witnesses to provide context, as aids to understanding and interpreting the portions that are cited, and/or as additional evidence of prior art status before the patent’s effective filing date, and/or as additional evidence of prior art status before the patent’s effective filing date. Sophos may also rely on uncited portions of Hoglund, other disclosed publications, and the testimony of experts to establish that a person of ordinary skill in the art would have been motivated to modify or combine certain of the cited references including Hoglund, so as to render the claims of the ’505 patent obvious.

² Plaintiffs appear in many instances to be pursuing overly broad constructions of limitations of the Asserted Claims in an effort to piece together an infringement claim where none exists. This claim chart takes into account Plaintiffs’ overly broad construction of the claim limitations. Any assertion that a particular limitation is disclosed by a prior art reference or references may be based on Plaintiffs’ apparent constructions and is not intended to be, and is not, an admission that such constructions are supportable or proper. Sophos reserves all rights to amend these contentions in view of any changes to Plaintiffs’ apparent constructions or claim construction positions, including as appearing in its Final Infringement Contentions, or in view of different or additional constructions rendered by the Court. Sophos is investigating this prior art and has not yet completed discovery from third parties, who may have relevant information concerning the prior art. Therefore, Sophos reserves the right to supplement this chart after additional discovery is received. To the extent that any of the prior art discloses the same or similar functionality or feature(s) of any of the accused products, Sophos reserves the right to argue that said feature or functionality does not practice any limitation of any of the Asserted Claims, and to argue, in the alternative, that if said feature or functionality is found to practice any limitation of any of the Asserted Claims, then the prior art reference teaches the limitation and that the claim is not patentable.

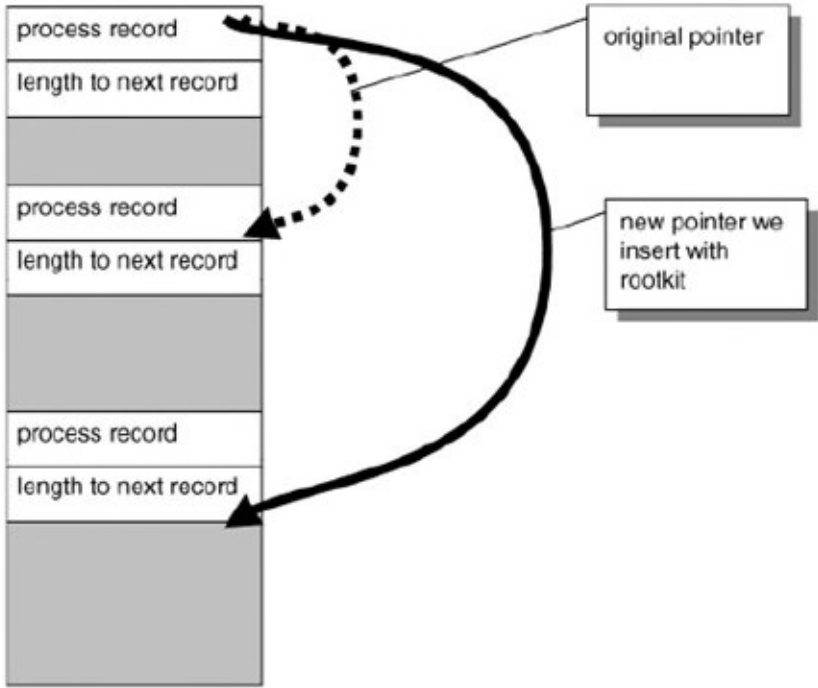
references, including but not limited to those found in the charts labeled Exhibit 505-A and Exhibit 505-B, in a manner that would have rendered the Asserted Claims invalid as obvious.

The citations in this chart are exemplary only. Any citation that refers to a figure or figure item incorporates by reference that figure or figure item, and any additional descriptions of that figure or figure item. Likewise, any citation to a figure incorporates by reference all descriptions of that figure. Sophos reserves the right to rely on the entirety of the references cited in this chart to show that the Asserted Claims are invalid. Citations presented for any claim limitation are expressly incorporated by reference into all other limitations for that claim as well as all limitations of all claims on which that claim depends. For every dependent claim, Sophos incorporates by reference all citations to all claims upon which that claim depends. Sophos also reserves the right to rely on additional citations or sources of evidence that also may be applicable, or that may become applicable in light of claim construction, changes in Plaintiffs' infringement contentions, and/or information obtained during discovery as the case progresses.

Claim Language	Exemplary Disclosures
[1P] A method for managing pestware on a computer comprising:	To the extent the preamble may be construed to be limiting, Hoglund discloses this limitation: “Any .sys file or executable that is to be loaded at boot time can be replaced, or the loader code can be inserted similarly to the way a virus can infect a file. Ironically, one of the best things to infect is a virus scanning or security product. A security product will typically start when the system is booted.” 38 “The rootkit can register itself as a driver which is loaded on boot. This requires creating a registry key. Again, the key can be hidden once the rootkit has loaded.” 38 “The kernel can be directly modified and saved to disk. A few changes must be made to the boot loader so that the kernel will pass a checksum integrity check. This can be very effective, since the kernel will be permanently modified, and no drivers will need to be registered.” 38 “Hoglund and Butler show exactly how to subvert the Windows XP and Windows 2000 kernels, teaching concepts that are easily applied to virtually any modern operating system, from Windows Server 2003 to Linux and UNIX. Using extensive downloadable examples, they teach rootkit programming techniques that can be used for a wide range of software, from white hat security tools to operating system drivers and debuggers. After reading this book, readers will be able to Understand the role of rootkits in remote command/control and software eavesdropping Build kernel rootkits that can make processes, files, and directories invisible Master key rootkit programming techniques, including hooking, runtime patching, and directly manipulating kernel objects Work with layered drivers to implement keyboard sniffers and file filters Detect rootkits and build host-based intrusion prevention software that resists rootkit attacks.” 2-3 Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.

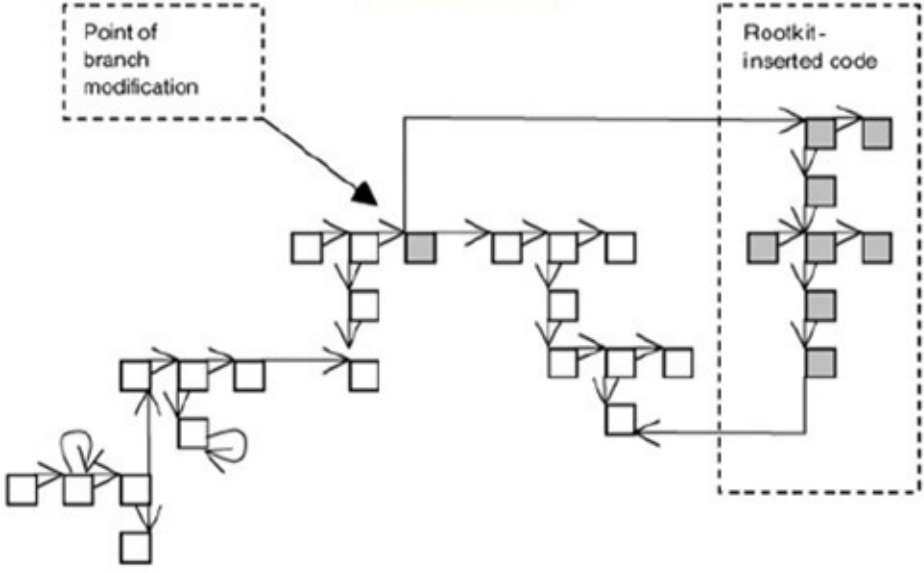
Claim Language	Exemplary Disclosures
[1A] monitoring events during a boot sequence of the computer;	Hoglund discloses this limitation. For example: “Any .sys file or executable that is to be loaded at boot time can be replaced, or the loader code can be inserted similarly to the way a virus can infect a file. Ironically, one of the best things to infect is a virus scanning or security product. A security product will typically start when the system is booted.” 38 “The rootkit can register itself as a driver which is loaded on boot. This requires creating a registry key. Again, the key can be hidden once the rootkit has loaded.” 38 “The kernel can be directly modified and saved to disk. A few changes must be made to the boot loader so that the kernel will pass a checksum integrity check. This can be very effective, since the kernel will be permanently modified, and no drivers will need to be registered.” 38 Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[1B] managing pestware-related events during a first period in a boot sequence of the computer, the first period in the boot sequence occurring before the computer becomes configured to run native applications, before a subsystem of an operating system is	Hoglund discloses this limitation. For example: “Any .sys file or executable that is to be loaded at boot time can be replaced, or the loader code can be inserted similarly to the way a virus can infect a file. Ironically, one of the best things to infect is a virus scanning or security product. A security product will typically start when the system is booted.” 38 “The rootkit can register itself as a driver which is loaded on boot. This requires creating a registry key. Again, the key can be hidden once the rootkit has loaded.” 38 “The kernel can be directly modified and saved to disk. A few changes must be made to the boot loader so that the kernel will pass a checksum integrity check. This can be very effective, since the kernel will be permanently modified, and no drivers will need to be registered.” 38

Claim Language	Exemplary Disclosures
loaded, and after a kernel is loaded;	Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[1C] managing pestware- related events in accordance with a set of behavior rules during a second period in the boot sequence occurring when the computer is configured to run native applications;	Hoglund discloses this limitation. For example: “Any .sys file or executable that is to be loaded at boot time can be replaced, or the loader code can be inserted similarly to the way a virus can infect a file. Ironically, one of the best things to infect is a virus scanning or security product. A security product will typically start when the system is booted.” 38 Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[1D] generating, in response to the monitoring, a record of events, the record of events including the pestware-related events;	Hoglund discloses this limitation. For example: “If detailed process tracking is enabled, for every process created an event will be recorded in the Event Log that looks something like that shown in Figure 7-6.” 249

Claim Language	Exemplary Disclosures
	Figure 4-5. Structure of SystemInformationClass buffer.  The information contained in the buffer comprises <code>_SYSTEM_PROCESSES</code> structures and their corresponding <code>_SYSTEM_THREADS</code> structures. One important item in the <code>_SYSTEM_PROCESSES</code> structure is the <code>UNICODE_STRING</code> containing the process name. There are also two <code>LARGE_INTEGER</code>s containing the user and kernel time used by the process. When you hide a process, your rootkit should add the time the process spent executing to another process in the list, so that all the recorded times add up to 100% of the CPU time.

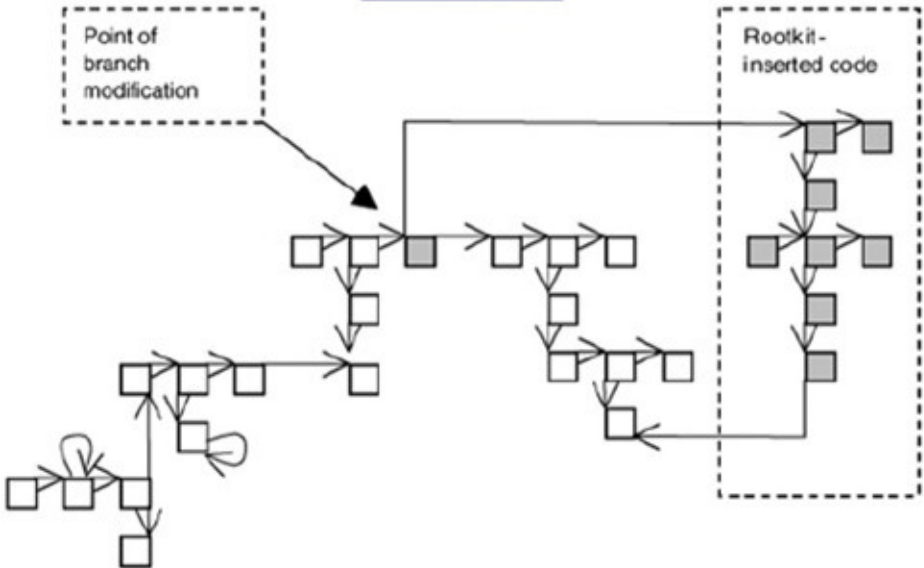
Claim Language	Exemplary Disclosures
	solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[1E] analyzing the record of events so as to identify the pestware-related events;	Hoglund discloses this limitation. For example: <i>See</i> Claim [1D]. “In the preceding code segment, pm_current is used to walk the list of loaded modules looking for the driver to hide, uni_hide_DriverName. For each module in the list, a comparison is made between the UNICODE strings of the driver to hide and the one currently being analyzed in the list.” 229 “Unique structure: The tools used for remote penetration should have structures and methods that are unique. Virus-detection solutions are always looking for the known. In virus-detection development, a publicly known virus is analyzed for general patterns, and these patterns are then applied to finding unknown viruses. If you attempt to download a rootkit from www.rootkit.com, for example, your virus scanner will likely quarantine the file. If they do not contain patterns found in known infections, then your tools will slip by undetected.” 283 “Still, let us assume you have discovered all the locations to watch in order to prevent rootkits from loading, and let's further assume you have resolved all the possible names of critical resources to protect. The difficulty you now face is in deciding when to signal. If you have detected a driver or a DLL loading, how do you know it is malware? Your detection software would need a signature for comparison, which assumes a known attack vector. Alternatively, your software could analyze the behavior of the module to try to determine whether it's malicious.” 347 Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.

Claim Language	Exemplary Disclosures
[1F] modifying the set of behavior rules so as to prevent the pestware related events; and	Hoglund discloses this limitation. For example: “In addition to memory-access restrictions, there are other security provisions. Some instructions are considered privileged, and can be used only in Ring Zero. These instructions are typically used to alter the behavior of the CPU or to directly access hardware. For example, the following x86 instructions are allowed only in Ring Zero.” 72 “Using layers, a developer can chain multiple drivers together. In this way, a developer can modify the behavior of an existing driver without coding a whole new driver from scratch.” 164 “In Chapter 4, we saw the power of using call hooks as a convenient way to modify program behavior. One downside of the call hook is that it modifies call tables, and this can be detected by anti-virus and anti-rootkit technology. A subtler approach to the problem is to patch the bytes within the function itself by inserting a jump into rootkit code. Additionally, modifying just a single function can affect multiple tables pointing to that function, without the need to keep track of all the tables that point to the function. This technique is called <i>detour patching</i>, and can be used to reroute the control flow around a function. Figure 5-1 illustrates how code is inserted by the rootkit into the control flow.” 139

Claim Language	Exemplary Disclosures
	Figure 5-1. Modification of control flow. View full size image  Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[1G] scanning data in a registry of the computer for pestware during the second	Hoglund discloses this limitation. For example: “The rootkit can register itself as a driver which is loaded on boot. This requires creating a registry key. Again, the key can be hidden once the rootkit has loaded.” 38

Claim Language	Exemplary Disclosures
period in the boot sequence.	“On Windows XP and greater, the memory pages containing the SSDT and IDT are set to read-only in the page table. If an attacker wishes to alter the contents of these memory pages, she must first change the pages to read/write. The best wayfor a rootkit to do this is called the CR0 trick, described later in this chapter. However, you can also make these tables writable by altering two registry keys.” 78-79 “In Windows NT/2000/XP/2003, there is a Registry key named HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\ Windows\AppInit_DLLs. Your rootkit can set the value of this key to one of its own DLLs that modifies the target process's IAT, or modifies kernel32.dll or ntdll.dll directly. When an application is loaded that uses User32.dll, the DLL listed as the value of this key will also be loaded by User32.dll into the application's address space... Your rootkit should hook whatever functions are its target if the rootkit DLL is being loaded for the first time by the process.” 96-97 “For example, you can load a rootkit into memory using Registry keys.” 347 “Finding all the ways a rootkit might be loaded is just the first step in defending against rootkits. Load-detection techniques are belabored by the need to decide both what to guard and when to signal. For example, you can load a rootkit into memory using Registry keys. An obvious detection point would be to hook ZwOpenKey, ZwCreateKey, and ZwSetValueKey (as did IPD). However, if your detection software hooks these functions, how does it know which keys to guard? Drivers are usually placed into the following key: HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services This key is a good location to filter in your Registry-hook function, but a rootkit could also alter another key: HKEY_LOCAL_MACHINE\System\ControlSet001\Services This key can be used when the machine is booted into the previously known good configuration.” 347 “Mark Russinovich and Bryce Cogswell have released a tool called Rootkit-Revealer.[7] It can detect hidden Registry entries as well as hidden files. To determine what the "truth" is, RootkitRevealer parses the files that correspond to the different Registry hives without the aide of the standard Win32 API calls, such as

Claim Language	Exemplary Disclosures
	RegOpenKeyEx and RegQueryValueEx. It also parses the file system at a very low level, avoiding the typical API calls. RootkitRevealer then calls the highest level APIs to compare the result with what it knows to be true. If a discrepancy is found, the behavior of the rootkit (and, hence, what it is hiding) is identified. This technique is fairly straightforward, yet very powerful.” 359 Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[2A] The method of claim 1, further comprising launching, after an operating system is initiated, a pestware management engine; and	Hoglund discloses this limitation. For example: “In addition to memory-access restrictions, there are other security provisions. Some instructions are considered privileged, and can be used only in Ring Zero. These instructions are typically used to alter the behavior of the CPU or to directly access hardware. For example, the following x86 instructions are allowed only in Ring Zero.” 72 “A typical module would include an entry point and perhaps a cleanup routine. For example, a Linux-loadable module may look something like this:” 44 “A secondary function of the OS is to provide debugging and diagnostic information about the system. For example, most operating systems can list the running or installed software. Most have logging mechanisms, so that applications can report when they have crashed, when someone fails to login properly, etc.” “Using layers, a developer can chain multiple drivers together. In this way, a developer can modify the behavior of an existing driver without coding a whole new driver from scratch.” 164 “In Chapter 4, we saw the power of using call hooks as a convenient way to modify program behavior. One downside of the call hook is that it modifies call tables, and this can be detected by anti-virus and anti-rootkit

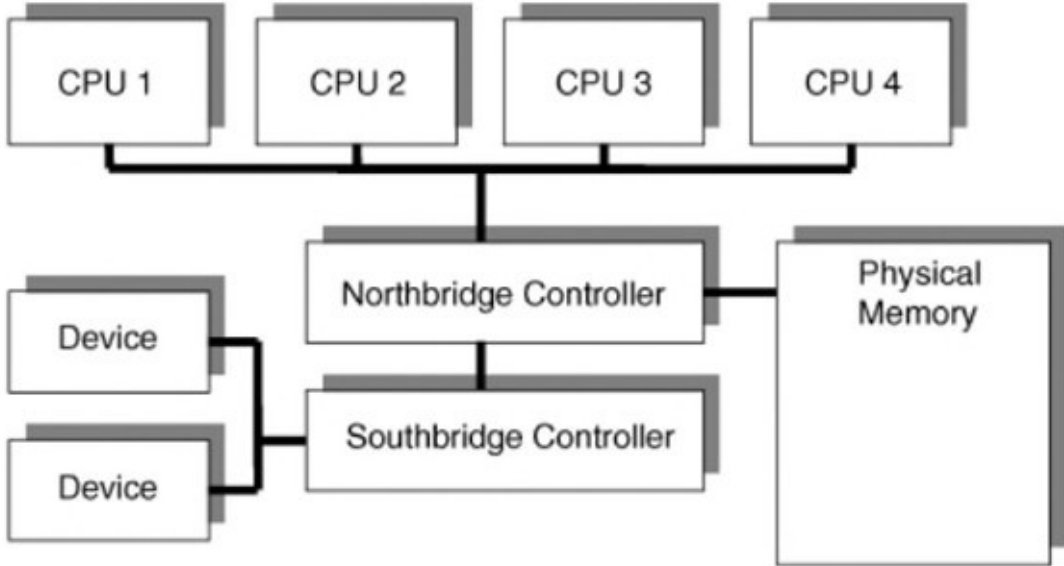
Claim Language	Exemplary Disclosures
	technology. A subtler approach to the problem is to patch the bytes within the function itself by inserting a jump into rootkit code. Additionally, modifying just a single function can affect multiple tables pointing to that function, without the need to keep track of all the tables that point to the function. This technique is called <i>detour patching</i>, and can be used to reroute the control flow around a function. Figure 5-1 illustrates how code is inserted by the rootkit into the control flow.” 139 Figure 5-1. Modification of control flow. View full size image  Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable

Claim Language	Exemplary Disclosures
	solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[2B] wherein the set of behavior rules includes behavior rules compiled by the pestware management engine.	Hoglund discloses this limitation. For example: <i>See Claims [1C], [1D], [2A].</i> Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[3] The method of claim 1, wherein the record of events includes information selected from a group consisting of: process identification information, file identification information, and hook generation information.	Hoglund discloses this limitation. For example: “When implementing an inline function hook, your rootkit will <i>actually overwrite the code bytes of the target function</i> so that no matter how or when the application resolves the function address, it will still be hooked.” 95 “In Chapter 4, we saw the power of using call hooks as a convenient way to modify program behavior. <i>One downside of the call hook is that it modifies call tables</i>, and this can be detected by anti-virus and anti-rootkit technology.” 139 Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.

Claim Language	Exemplary Disclosures
[4] The method of claim 1, further comprising: managing pestware-related events after the computer becomes configured to run native applications	Hoglund discloses this limitation. For example: <i>See Claims [1A], [1C], [1D], [1G].</i> Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[5] The method of claim 1, wherein the monitoring includes monitoring the boot sequence while boot drivers are initiated.	Hoglund discloses this limitation. For example: <i>See Claims [1C], [1E].</i> “Any .sys file or executable that is to be loaded at boot time can be replaced, or the loader code can be inserted similarly to the way a virus can infect a file. Ironically, one of the best things to infect is a virus scanning or security product. A security product will typically start when the system is booted.” 38 “The rootkit can register itself as a driver which is loaded on boot. This requires creating a registry key. Again, the key can be hidden once the rootkit has loaded.” 38 “The kernel can be directly modified and saved to disk. A few changes must be made to the boot loader so that the kernel will pass a checksum integrity check. This can be very effective, since the kernel will be permanently modified, and no drivers will need to be registered.” 38 Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable

Claim Language	Exemplary Disclosures
	solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[6P] A system for managing pestware on a computer comprising:	To the extent the preamble may be construed to be limiting, Hoglund discloses this limitation. For example: <i>See Claim [1P].</i> Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[6A] a processor;	Hoglund discloses this limitation. For example: “We begin our discussion of hardware mechanisms by taking a look at how the Intel x86 family of microprocessors performs access control. We then discuss how the processor keeps track of matters using lookup tables. We also discuss control registers and, more importantly, how memory pages work.” 70

Claim Language	Exemplary Disclosures
	The diagram illustrates a memory protection architecture using four concentric circles. The innermost circle is labeled 'Ring 0'. The outermost circle is labeled 'Ring 3'. A dot is placed within the 'Ring 0' circle, with a line pointing to it from the label 'Kernel Programs'. Another dot is placed in the annular region between the second and third circles from the center, with a line pointing to it from the label 'User Programs'. In addition to memory-access restrictions, there are other security provisions. Some instructions are considered privileged, and can be used only in Ring Zero. These instructions are typically used to alter the behavior of the CPU or to directly access hardware. For example, the following x86 instructions are allowed only in Ring Zero: Figure 3-1

Claim Language	Exemplary Disclosures
	Multiple-processor systems are like multi-threaded environments in a way, because code can be executing on two or more CPUs at once. Chapter 7, Direct Kernel Object Manipulation, covers multiprocessor synchronization. The layout of a typical multiprocessor system is shown in Figure 3-8. As the figure illustrates, multiple CPUs share access to a single memory area, set of controllers, and group of devices. Figure 3-8. A typical multiprocessor bus layout.  Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable

Claim Language	Exemplary Disclosures
	solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[6B] a memory comprising:	Hoglund discloses this limitation. For example: <i>See</i> Claim [6A]. Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[6C] a first instruction set configured to monitor events during a boot sequence of the computer;	Hoglund discloses this limitation. For example: <i>See</i> Claim [1A]. Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[6D] a second instruction set configured to manage pestware-related events during first period in a boot sequence of the computer, the first	Hoglund discloses this limitation. For example: <i>See</i> Claim [1B]. Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come

Claim Language	Exemplary Disclosures
period in the boot sequence occurring before the computer becomes configured to run native applications, before a WIN32 subsystem is loaded, and after a kernel is loaded;	from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[6E] a third instruction set configured to manage pestware-related events in accordance with a set of behavior rules during a second period in the boot sequence occurring when the computer is configured to run native applications;	Hoglund discloses this limitation. For example: <i>See Claim [1C].</i> Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[6F] a fourth instruction set configured to generate, in response to the monitoring, a record of events, the record of events including the	Hoglund discloses this limitation. For example: <i>See Claim [1D].</i> Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable

Claim Language	Exemplary Disclosures
pestware-related events;	solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[6G] a fifth instruction set configured to analyze the record of events so as to identify the pestware-related events; and	Hoglund discloses this limitation. For example: <i>See Claim [1E].</i> Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[6H] a sixth instruction set configured to modify the set of behavior rules so as to prevent the pestware related events; and	Hoglund discloses this limitation. For example: <i>See Claims [1E], [1F].</i> Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[6I] a seventh instruction set configured to scan data in a registry of the computer for pestware during the second	Hoglund discloses this limitation. For example: <i>See Claim [1G].</i> Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come

Claim Language	Exemplary Disclosures
period in the boot sequence.	from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[7A] The system of claim 6, wherein the memory further comprises an eighth instruction set configured to launch, after an operating system is initiated, a pestware management engine; and	Hoglund discloses this limitation. For example: <i>See Claim [2A].</i> Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[7B] wherein the set of behavior rules includes behavior rules compiled by the pestware management engine.	Hoglund discloses this limitation. For example: <i>See Claim [2B].</i> Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[8] The system of claim 1, wherein the record of events includes information selected from a group consisting	Hoglund discloses this limitation. For example: <i>See Claim [3].</i> Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or

Claim Language	Exemplary Disclosures
of: process identification information, file identification information, and hook generation information.	when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[9] The system of claim 6, wherein the memory further comprises: a ninth instruction set configured to manage pestware-related events after the computer becomes configured to run native applications.	Hoglund discloses this limitation. For example: <i>See</i> Claims [1P], [1A], [1C], [1G]. Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[10] The system of claim 6, wherein the first instruction set is further configured to monitor the boot sequence while boot drivers are initiated	Hoglund discloses this limitation. For example: <i>See</i> Claims [1E], [5]. Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[11P] A non-transitory tangible computer-readable storage	To the extent the preamble may be construed to be limiting, Hoglund discloses this limitation. For example: <i>See</i> Claims [1P], [6P].

Claim Language	Exemplary Disclosures
medium comprising program instructions for:	Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[11A] monitoring events during a boot sequence of a computer;	Hoglund discloses this limitation. For example: <i>See Claim [1A].</i> Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[11B] managing pestware-related events during a first period in a boot sequence of the computer, the first period in the boot sequence occurring before the computer becomes configured to run native applications, before a WIN32 subsystem is loaded,	Hoglund discloses this limitation. For example: <i>See Claim [1B].</i> Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.

Claim Language	Exemplary Disclosures
and after a kernel is loaded;	
[11C] managing pestware-related events in accordance with a set of behavior rules during a second period in the boot sequence occurring when the computer is configured to run native applications;	Hoglund discloses this limitation. For example: <i>See Claim [1C].</i> Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[11D] generating, in response to the monitoring, a record of events, the record of events including the pestware-related events;	Hoglund discloses this limitation. For example: <i>See Claim [1D].</i> Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[11E] analyzing the record of events so as to identify the pestware-related events;	Hoglund discloses this limitation. For example: <i>See Claim [1E].</i> Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come

Claim Language	Exemplary Disclosures
	from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[11F] modifying the set of behavior rules so as to prevent the pestware related events; and	Hoglund discloses this limitation. For example: <i>See Claim [1F].</i> Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[11G] scanning data in a registry of the computer for pestware during the second period in the boot sequence.	Hoglund discloses this limitation. For example: <i>See Claim [1G].</i> Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[12A] The non-transitory tangible computer-readable storage medium of claim 11 further comprising program	Hoglund discloses this limitation. For example: <i>See Claim [2A].</i> Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or

Claim Language	Exemplary Disclosures
instructions for launching, after an operating system is initiated, a pestware management engine; and	when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[12B] wherein the set of behavior rules includes behavior rules compiled by the pestware management engine.	Hoglund discloses this limitation. For example: <i>See Claim [2B].</i> Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[13] The non-transitory tangible computer-readable storage medium of claim 11, wherein the record of events includes information selected from a group consisting of: process identification information, file identification	Hoglund discloses this limitation. For example: <i>See Claim [3].</i> Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.

Claim Language	Exemplary Disclosures
information, and hook generation information.	
[14] The non-transitory tangible computer-readable storage medium of claim 11 further comprising program instructions for: managing pestware-related events after the computer becomes configured to run native applications.	Hoglund discloses this limitation. For example: <i>See Claim [4].</i> Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[15] The non-transitory tangible computer-readable storage medium of claim 11, wherein the monitoring includes monitoring the boot sequence while boot drivers are initiated.	Hoglund discloses this limitation. For example: <i>See Claims [1E], [5].</i> Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[16] The method of claim 2, further comprising modifying, by the pestware management engine, the behavior rules	Hoglund discloses this limitation. For example: <i>See Claim [1F].</i> Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or

Claim Language	Exemplary Disclosures
utilized by a kernel-level monitor, wherein the kernel level monitor manages pestware-related events in the first and second periods in the boot sequence.	when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[17] The method of claim 1, further comprising modifying the set of behavior rules based on the scanned registry so as to prevent the pestware-related events.	Hoglund discloses this limitation. For example: <i>See Claims [1F], [1G].</i> “The rootkit can register itself as a driver which is loaded on boot. This requires creating a registry key. Again, the key can be hidden once the rootkit has loaded.” 38 “On Windows XP and greater, the memory pages containing the SSDT and IDT are set to read-only in the page table. If an attacker wishes to alter the contents of these memory pages, she must first change the pages to read/write. The best wayfor a rootkit to do this is called the CR0 trick, described later in this chapter. However, you can also make these tables writable by altering two registry keys.” 78-79 “In Windows NT/2000/XP/2003, there is a Registry key named HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\ Windows\AppInit_DLLs. Your rootkit can set the value of this key to one of its own DLLs that modifies the target process's IAT, or modifies kernel32.dll or ntdll.dll directly. When an application is loaded that uses User32.dll, the DLL listed as the value of this key will also be loaded by User32.dll into the application's address space... Your rootkit should hook whatever functions are its target if the rootkit DLL is being loaded for the first time by the process.” 96-97 “For example, you can load a rootkit into memory using Registry keys.” 347 “Finding all the ways a rootkit might be loaded is just the first step in defending against rootkits. Load-detection techniques are belabored by the need to decide both what to guard and when to signal. For example,

Claim Language	Exemplary Disclosures
	you can load a rootkit into memory using Registry keys. <i>An obvious detection point would be to hook ZwOpenKey, ZwCreateKey, and ZwSetValueKey (as did IPD). However, if your detection software hooks these functions, how does it know which keys to guard?</i> Drivers are usually placed into the following key: HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services This key is a good location to filter in your Registry-hook function, but a rootkit could also alter another key: HKEY_LOCAL_MACHINE\System\ControlSet001\Services This key can be used when the machine is booted into the previously known good configuration.” 347 “Mark Russinovich and Bryce Cogswell have released a tool called Rootkit-Revealer.[7] It can detect hidden Registry entries as well as hidden files. To determine what the "truth" is, RootkitRevealer parses the files that correspond to the different Registry hives without the aide of the standard Win32 API calls, such as RegOpenKeyEx and RegQueryValueEx. It also parses the file system at a very low level, avoiding the typical API calls. RootkitRevealer then calls the highest level APIs to compare the result with what it knows to be true. If a discrepancy is found, the behavior of the rootkit (and, hence, what it is hiding) is identified. This technique is fairly straightforward, yet very powerful.” 359 Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[18] The method of claim 1, wherein the pestware-related events include events associated with at least	Hoglund discloses this limitation. For example: “In addition to memory-access restrictions, there are other security provisions. Some instructions are considered privileged, and can be used only in Ring Zero. These instructions are typically used to alter the

Claim Language	Exemplary Disclosures
one of loading pestware, executing pestware, and one or more behaviors of pestware.	behavior of the CPU or to directly access hardware. For example, the following x86 instructions are allowed only in Ring Zero.” 72 “Using layers, a developer can chain multiple drivers together. In this way, a developer can modify the behavior of an existing driver without coding a whole new driver from scratch.” 164 “In Chapter 4, we saw the power of using call hooks as a convenient way to modify program behavior. One downside of the call hook is that it modifies call tables, and this can be detected by anti-virus and anti-rootkit technology. A subtler approach to the problem is to patch the bytes within the function itself by inserting a jump into rootkit code. Additionally, modifying just a single function can affect multiple tables pointing to that function, without the need to keep track of all the tables that point to the function. This technique is called <i>detour patching</i>, and can be used to reroute the control flow around a function. Figure 5-1 illustrates how code is inserted by the rootkit into the control flow.” 139 “Associate an Endpoint with a Local Address: Having created both an endpoint object and a local address object, our next step is to associate them. An endpoint is worthless without an associated address. The address tells the system which local port and IP address you wish to use. In our example, we have configured the address so that the system will choose a local port for us (similar to the way you expect a socket to work). Communication with the underlying driver will take place using IOCTL IRPs from this point forward. For each function we wish to call, we must first craft an IRP, fill it with arguments and data, and then pass it down to the next-lowest driver via the IoCallDriver() routine. After we pass each IRP, we must wait for it to complete. To do this, we use a completion routine. An event shared between the completion routine and the rest of our code allows us to wait for processing to complete.” 296 Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.

Claim Language	Exemplary Disclosures
[19] The system of claim 6, wherein the pestware-related invents include events associated with at least one of loading pestware, executing pestware, and one or more behaviors of pestware.	Hoglund discloses this limitation. For example: <i>See Claim [18].</i> Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.
[20] The non-transitory tangible computer-readable storage medium of claim 11, wherein the pestware-related invents include events associated with at least one of loading pestware, executing pestware, and one or more behaviors of pestware.	Hoglund discloses this limitation. For example: <i>See Claim [18].</i> Furthermore, this claim element is obvious in light of Hoglund itself, when combined with any of the other references as charted for this claim element in the charts labeled Exhibit 505-A and Exhibit 505-B, and/or when combined with the knowledge of one of ordinary skill in the art. Motivations to combine may come from the knowledge of the person of ordinary skill themselves, or from the known problems and predictable solutions as embodied in these references. Further motivations to combine references and additional details may be found in the cover pleading and Exhibit 505-B.