

SOPHOS FINAL INVALIDITY CONTENTIONS

EXHIBIT 505-A-21

Prior Art Reference: A Virtual Machine Introspection Based Architecture for Intrusion Detection (“Garfinkel”)

Garfinkel was available at least as of 2003 and therefore qualifies as prior art under at least 35 U.S.C. §§ 102(a), (b), (e) and/or 103 as to claims 1-20 (the “Asserted Claims”) of U.S. Pat. No. 8,856,505 (“the ’505 patent”).¹ Garfinkel, including any material incorporated by reference into Garfinkel, 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 Garfinkel 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 Garfinkel and/or to combine the teachings of Garfinkel with other prior art references, including but not limited to any systems relating to Garfinkel or any other

¹ Sophos may rely upon uncited portions of Garfinkel, 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 Garfinkel, 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 Garfinkel, 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.

prior art 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, Garfinkel discloses this limitation: “In this paper we present a new architecture for building intrusion detection systems that provides good visibility into the state of the monitored host, while still providing strong isolation for the IDS, thus lending significant resistance to both evasion and attack.” Abstract “We propose the idea of virtual machine introspection, an approach to intrusion detection which co-locates an IDS on the same machine as the host it is monitoring and leverages a virtual machine monitor to isolate the IDS from the monitored host. The activity of the host is analyzed by directly observing hardware state and inferring software state based on a priori knowledge of its structure. This approach allows the IDS to: maintain high visibility, provides high evasion resistance in the face of host compromise, provides high attack resistance due to strong isolation, and provides the unique capability to mediate access to host hardware, allowing hardware access control policies to be enforced in the face of total host compromise.” 14 “At the heart of any intrusion detection system is the policy engine. This component interprets system state and events from the VMM interface and OS interface library, and decides whether or not the system has been compromised. If the system has been compromised, the policy engine is responsible for responding in an appropriate manner. For example, in case of a break-in, the policy engine can suspend or reboot the virtual machine, and report the break-in.” 7

Claim Language

Exemplary Disclosures

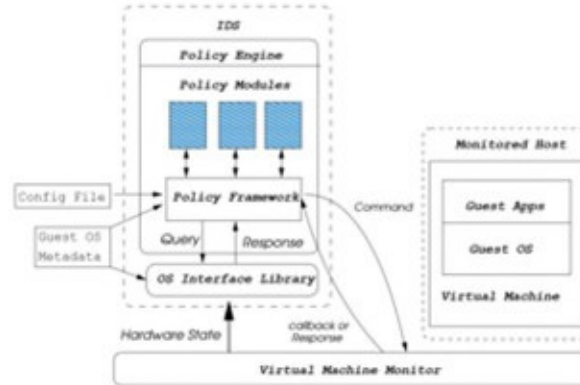


Figure 1. A High-Level View of our VMI-Based IDS Architecture: On the right is the virtual machine (VM) that runs the host being monitored. On the left is the VMI-based IDS with its major components: the OS interface library that provides an OS-level view of the VM by interpreting the hardware state exported by the VMM, the policy engine consisting of a common framework for building policies, and policy modules that implement specific intrusion detection policies. The virtual machine monitor provides a substrate that isolates the IDS from the monitored VM and allows the IDS to inspect the state of the VM. The VMM also allows the IDS to interpose on interactions between the guest OS/guest applications and the virtual hardware.

Name	Description
cdoor	Stealth user level remote backdoor
t0rn	Precompiled user level rootkit
Ramen	Linux Worm
lrk5	Source based user level rootkit
knark-0.59	LKM based kernel backdoor/rootkit
adore-0.42	LKM based kernel backdoor/rootkit
dsniff 2.4	All-purpose packet sniffer for switched networks
SUCKIT	/dev/kmem patching based kernel backdoor

Table 1

Claim Language	Exemplary Disclosures
	Furthermore, this claim element is obvious in light of Garfinkel 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.
[1A] monitoring events during a boot sequence of the computer;	Garfinkel discloses this limitation. For example: “Polling modules periodically check the system for signs of malicious activity... Event-driven checkers run when the VMM detects changes to hardware state, such as a write to a sensitive location in memory.” 8-9 “Detecting tampering with an OS code segment can be an [sic] useful mechanism for discovering the presence of malicious code, and preventing its installation into the kernel proper. Our kernel memory enforcer works by marking the code section, sys call table, and other sensitive portions of the kernel as read-only through the VMM. If a malicious program, such as a kernel back door tries to modify these sections of memory, the VM will be halted and the kernel memory protection enforcer notified.” 10 “Our Memory Protection Enforcer (“mem”) was able to detect and prevent the installation of all of our kernel backdoors. knark and adore were stopped by blocking their attempt to modify sys call table. SUCKIT was stopped by blocking its attempt to modify the interrupt dispatch table.” 10 Furthermore, this claim element is obvious in light of Garfinkel 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
[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 loaded, and after a kernel is loaded;	Garfinkel discloses this limitation. For example: “Our Memory Protection Enforcer (“mem”) was able to detect and prevent the installation of all of our kernel backdoors. knark and adore were stopped by blocking their attempt to modify sys call table. SUCKIT was stopped by blocking its attempt to modify the interrupt dispatch table.” 10 Furthermore, this claim element is obvious in light of Garfinkel 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;	Garfinkel discloses this limitation. For example: “The lie detector module works by directly inspecting hardware and kernel state, and by querying the host system through user-level programs (e.g. ps, ifconfig, netstat) via a remote shell. If it detects conflicts between these two views (i.e. the system is lying), it reports the presence of malicious tampering.” 9 “The raw socket detector is a “burglar alarm” [37] style policy module for detecting the use of raw sockets by user-level programs for the purpose of catching such malicious applications. This is accomplished by querying the kernel about the type of all sockets held by user processes.” 9 “At the heart of any intrusion detection system is the policy engine. This component interprets system state and events from the VMM interface and OS interface library, and decides whether or not the system has been compromised. If the system has been compromised, the policy engine is responsible for responding in an appropriate manner. For example, in case of a break-in, the policy engine can suspend or reboot the virtual machine, and report the breakin. Since the focus of our work has been studying VMI as a platform for IDS, we have focused on implementing variations on mainstream HIDS style policies [37] such as

Claim Language	Exemplary Disclosures																																																																								
	burglar alarms, misuse detectors and integrity checkers. A policy engine implementing complex anomaly detection and other, more exotic techniques can also be supported in this architecture.” 7 “Detecting tampering with an OS code segment can be an useful mechanism for discovering the presence of malicious code, and preventing its installation into the kernel proper. Our kernel memory enforcer works by marking the code section, sys_call_table, and other sensitive portions of the kernel as read-only through the VMM. If a malicious program, such as a kernel back door tries to modify these sections of memory, the VM will be halted and the kernel memory protection enforcer notified.” 10 “The NIC Access Enforcer prevents the Ethernet device entering promiscuous mode, or being configured with a MAC address which has not been pre-specified. Using this module we can prevent variety of common misuses of the NIC to be detected and prevented. In spite of its simple functionality the NIC module provides a useful policy enforcement tool. It is more robust to attack than normal host-based solutions, and not susceptible to evasion, as is a problem with remote promiscuous mode detection solutions [9].” 10 <table><tr><th>Name</th><th>Description</th><th>nic</th><th>raw</th><th>sig</th><th>int</th><th>lie</th><th>mem</th></tr><tr><td>cdoor</td><td>Stealth user level remote backdoor</td><td></td><td>D</td><td></td><td></td><td></td><td></td></tr><tr><td>t0rn</td><td>Precompiled user level rootkit</td><td></td><td></td><td>D</td><td></td><td>D</td><td></td></tr><tr><td>Ramen</td><td>Linux Worm</td><td></td><td></td><td>D</td><td></td><td></td><td></td></tr><tr><td>lrk5</td><td>Source based user level rootkit</td><td>P</td><td></td><td>D</td><td>D</td><td>D</td><td></td></tr><tr><td>knark-0.59</td><td>LKM based kernel backdoor/rootkit</td><td></td><td></td><td>D</td><td></td><td>D</td><td>P</td></tr><tr><td>adore-0.42</td><td>LKM based kernel backdoor/rootkit</td><td></td><td></td><td>D</td><td></td><td>D</td><td>P</td></tr><tr><td>dsniff 2.4</td><td>All-purpose packet sniffer for switched networks</td><td>P</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>SUCKIT</td><td>/dev/kmem patching based kernel backdoor</td><td></td><td></td><td>D</td><td></td><td>D</td><td>P</td></tr></table> Table 1. Results of Livewire policy modules against common attacks. Within the grid, “P” designates a prevented attack and “D” a detected attack.	Name	Description	nic	raw	sig	int	lie	mem	cdoor	Stealth user level remote backdoor		D					t0rn	Precompiled user level rootkit			D		D		Ramen	Linux Worm			D				lrk5	Source based user level rootkit	P		D	D	D		knark-0.59	LKM based kernel backdoor/rootkit			D		D	P	adore-0.42	LKM based kernel backdoor/rootkit			D		D	P	dsniff 2.4	All-purpose packet sniffer for switched networks	P						SUCKIT	/dev/kmem patching based kernel backdoor			D		D	P
Name	Description	nic	raw	sig	int	lie	mem																																																																		
cdoor	Stealth user level remote backdoor		D																																																																						
t0rn	Precompiled user level rootkit			D		D																																																																			
Ramen	Linux Worm			D																																																																					
lrk5	Source based user level rootkit	P		D	D	D																																																																			
knark-0.59	LKM based kernel backdoor/rootkit			D		D	P																																																																		
adore-0.42	LKM based kernel backdoor/rootkit			D		D	P																																																																		
dsniff 2.4	All-purpose packet sniffer for switched networks	P																																																																							
SUCKIT	/dev/kmem patching based kernel backdoor			D		D	P																																																																		
	Furthermore, this claim element is obvious in light of Garfinkel 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																																																																								

Claim Language	Exemplary Disclosures
	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;	Garfinkel discloses this limitation. For example: “Polling modules periodically check the system for signs of malicious activity... Event-driven checkers run when the VMM detects changes to hardware state, such as a write to a sensitive location in memory.” 8-9 ““At startup, each event-driven checker registers all of the events it would like to be notified of with the policy framework.” 9 “However, the cost of this was that monitoring frequent asynchronous events, e.g. all system calls, may be quite performance intensive. Our current architecture could support frequent asynchronous checks, such as monitoring and processing system call, and supporting lightweight data watchpoints with relative efficiency via. hard coding the functionality <i>to log these events directly into the monitor, then offloading the processing of these logs to the policy engine.</i>” 13 Furthermore, this claim element is obvious in light of Garfinkel 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.
[1E] analyzing the record of events so as to identify the pestware-related events;	Garfinkel discloses this limitation. For example: “Polling modules periodically check the system for signs of malicious activity... Event-driven checkers run when the VMM detects changes to hardware state, such as a write to a sensitive location in memory.” 8-9 “At startup, each event-driven checker registers all of the events it would like to be notified of with the policy framework.” 9

Claim Language	Exemplary Disclosures
	“Detecting tampering with an OS code segment can be an [sic] useful mechanism for discovering the presence of malicious code, and preventing its installation into the kernel proper. Our kernel memory enforcer works by marking the code section, sys call table, and other sensitive portions of the kernel as read-only through the VMM. If a malicious program, such as a kernel back door tries to modify these sections of memory, the VM will be halted and the kernel memory protection enforcer notified.” 10 “The NIC Access Enforcer prevents the Ethernet device entering promiscuous mode, or being configured with a MAC address which has not been pre-specified. Using this module we can prevent variety of common misuses of the NIC to be detected and prevented.” 10 “[o]ur current architecture could support frequent asynchronous checks, such as monitoring and processing system call, and supporting lightweight data watchpoints with relative efficiency via. hard coding the functionality <i>to log these events directly into the monitor, then offloading the processing of these logs to the policy engine.</i>” 13 Furthermore, this claim element is obvious in light of Garfinkel 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.
[1F] modifying the set of behavior rules so as to prevent the pestware related events; and	Garfinkel discloses this limitation. For example: “Each policy module is an individual Python module (i.e. a single file) that leverages the policy framework. Policy modules can be run stand-alone or in concert with other policy modules.” 8 “LIVEWIRE FRONT END: The front end code is responsible for bootstrapping the system, starting the OS interface library process, loading policy modules, and running policy modules in concert. Interfaces are provided for obtaining configuration information, reporting intrusions, and registering policy modules with a common controller.

Claim Language	Exemplary Disclosures																																																																								
	We have implemented six sample security policy modules in Livewire. Four modules are polling modules, modules that run periodically and check for signs of an intrusion. The other two are event-driven modules that are triggered by a specific event, such as an attempt to write to sensitive memory.” 8 “Our Signature Scanner (“sig”) was able to detect all of the attacks we provided signatures for.... Our Raw Socket Detector (“raw”) raw socket detector detected the presence of cd00r and a similar stealth backdoor we included in our test attack suite. Our Lie Detector (“lie”) modules proved especially effective against rootkits both at kernel and user level since they all provided functionality to hide processes. The User Program Integrity Checker (“int”) was able to detect the presence of t0rn and lrk5 based on their use of backdoored version of inetd and sshd. We also found it also effective in detecting backdoored versions of syslog. Our Memory Protection Enforcer (“mem”) was able to detect and prevent the installation of all of kernel backdoors. knark and adore were stopped by blocking their attempt to modify sys call table. SUCKIT stopped by blocking its attempt to modify the interrupt dispatch table. Our NIC access enforcer (“nic”) was trivially able to detect and prevent the packet sniffers in our test attack suite from operating, based on their reliance on running the NIC in promiscuous mode.” 10-11 “There are many sensitive mutable kernel data structures that we do not yet protect that could present an avenue for attack. We have simply tried to “raise the bar,” and prevent the most obvious of cases of this class of attack Finding better methods for identifying and enforcing the static and dynamic invariants that a VMI IDS relies upon seems an important area for further study.” 12 <table><tr><th>Name</th><th>Description</th><th>nic</th><th>raw</th><th>sig</th><th>int</th><th>lie</th><th>mem</th></tr><tr><td>cd00r</td><td>Stealth user level remote backdoor</td><td></td><td>D</td><td></td><td></td><td></td><td></td></tr><tr><td>rtm</td><td>Precompiled user level rootkit</td><td></td><td></td><td>D</td><td></td><td></td><td>D</td></tr><tr><td>Ramon</td><td>Linux Worm</td><td></td><td></td><td>D</td><td></td><td></td><td></td></tr><tr><td>sk7</td><td>Source based user level rootkit</td><td>P</td><td></td><td>D</td><td>D</td><td>D</td><td></td></tr><tr><td>knark-0.56</td><td>LKM based kernel backdoor/rootkit</td><td></td><td></td><td>D</td><td></td><td>D</td><td>P</td></tr><tr><td>adore-0.42</td><td>LKM based kernel backdoor/rootkit</td><td></td><td></td><td>D</td><td></td><td>D</td><td>P</td></tr><tr><td>sniff 2.4</td><td>All-purpose packet sniffer for switched networks</td><td>P</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>SUCKIT</td><td>udev kernel patching based kernel backdoor</td><td></td><td></td><td>D</td><td></td><td>D</td><td>P</td></tr></table> Table 1. Results of Livewire policy modules against common attacks. Within the grid, “P” designates a prevented attack and “D” a detected attack. Furthermore, this claim element is obvious in light of Garfinkel 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	Name	Description	nic	raw	sig	int	lie	mem	cd00r	Stealth user level remote backdoor		D					rtm	Precompiled user level rootkit			D			D	Ramon	Linux Worm			D				sk7	Source based user level rootkit	P		D	D	D		knark-0.56	LKM based kernel backdoor/rootkit			D		D	P	adore-0.42	LKM based kernel backdoor/rootkit			D		D	P	sniff 2.4	All-purpose packet sniffer for switched networks	P						SUCKIT	udev kernel patching based kernel backdoor			D		D	P
Name	Description	nic	raw	sig	int	lie	mem																																																																		
cd00r	Stealth user level remote backdoor		D																																																																						
rtm	Precompiled user level rootkit			D			D																																																																		
Ramon	Linux Worm			D																																																																					
sk7	Source based user level rootkit	P		D	D	D																																																																			
knark-0.56	LKM based kernel backdoor/rootkit			D		D	P																																																																		
adore-0.42	LKM based kernel backdoor/rootkit			D		D	P																																																																		
sniff 2.4	All-purpose packet sniffer for switched networks	P																																																																							
SUCKIT	udev kernel patching based kernel backdoor			D		D	P																																																																		

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.
[1G] scanning data in a registry of the computer for pestware during the second period in the boot sequence.	Garfinkel discloses this limitation. For example: “Scanning the file system for the presence of known malicious program based on a known “signature” substring of the program is a popular intrusion detection technique. It is employed by anti-virus tools as well as root-kit detection tools like chkrootkit [31]. These tools leverage the fact that most attackers do not write their own tools, but instead rely on a relatively small number of publicly available rootkits, backdoors, Trojan horses and other attack tools. Popular examples include “subseven,” “backorifice,” and “netbus” Trojan horses for Windows, or the “adore” and “knark” kernel backdoors under Linux.” 9 “Our Signature Scanner (“sig”) was able to detect all of the attacks we provided signatures for. This was a predictable result; the only interesting surprise was that it often detected the presence of malicious code before it had been run, based on the presence of the signature in the buffer cache (when we examined them on disk), in the memory of ssh(as they were being downloaded), or in memory once activated.” 10 Furthermore, this claim element is obvious in light of Garfinkel 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	Garfinkel discloses this limitation. For example: “At the heart of any intrusion detection system is the policy engine. This component interprets system state and events from the VMM interface and OS interface library, and decides whether or not the system has been compromised. If the system has been compromised, the policy engine is responsible for responding in an appropriate manner. For example, in case of a break-in, the policy engine can suspend or reboot the virtual machine, and report the breakin. Since the focus of our work has been studying VMI as a platform for IDS,

Claim Language	Exemplary Disclosures
pestware management engine; and	we have focused on implementing variations on mainstream HIDS style policies [37] such as burglar alarms, misuse detectors and integrity checkers. A policy engine implementing complex anomaly detection and other, more exotic techniques can also be supported in this architecture.” 7 <pre> graph TD subgraph PESTWARE_MANAGEMENT_ENGINE [PESTWARE MANAGEMENT ENGINE] subgraph Policy_Engine [Policy Engine] subgraph Policy_Modules [Policy Modules] direction TB PM1[] PM2[] PM3[] end Policy_Framework[Policy Framework] end OS_Interface_Library[OS Interface Library] Policy_Engine <--> OS_Interface_Library end Config_File[Config File] --> Policy_Framework Guest_OS_Metadata[Guest OS Metadata] --> OS_Interface_Library Hardware_State[Hardware State] --> OS_Interface_Library Policy_Framework -- Query --> OS_Interface_Library OS_Interface_Library -- Response --> Policy_Framework OS_Interface_Library -- Command --> Monitored_Host subgraph Monitored_Host [Monitored Host] Guest_Apps[Guest Apps] Guest_OS[Guest OS] end subgraph Virtual_Machine [Virtual Machine] Monitored_Host end Virtual_Machine_Monitor[Virtual Machine Monitor] -- "callback or response" --> OS_Interface_Library </pre> Figure 1. “The policy engine consists of two pieces: the policy framework, a common API for writing security policies, and the policy modules that implement actual security policies. The policy engine was built entirely using Python... We have implemented six sample security policy modules in Livewire. Four modules are polling modules, modules that run periodically and check for signs of an intrusion. The other two are event-driven modules that are triggered by a specific event, such as an attempt to write to sensitive memory.” 8 Furthermore, this claim element is obvious in light of Garfinkel 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.
[2B] wherein the set of behavior rules includes behavior rules compiled by the pestware management engine.	Garfinkel discloses this limitation. For example: <i>See Claims [1C], [1D], [2A].</i> Furthermore, this claim element is obvious in light of Garfinkel 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.	Garfinkel discloses this limitation. For example: “The raw socket detector is a “burglar alarm” [37] style policy module for detecting the use of raw sockets by user-level programs for the purpose of catching such malicious applications. This is accomplished by querying the kernel about the type of all sockets held by user processes.” 9 “Our signature detector performs a scan of all of host memory for attack signatures. This more aggressive approach requires a more careful selection of signatures to avoid false positives. It also means that malicious programs that have not yet been installed may also be detected, e.g. <i>in the filesystem buffer cache.</i>” 9 “Other IDS tools can benefit from the capability of a VMM to allow secure collocation of monitoring on the same machine as the host, even without the use of introspection. HIDS techniques such as filesystem integrity checking could easily be moved outside of the host for better isolation.” 14 “Detecting tampering with an OS code segment can be an useful mechanism for discovering the presence of malicious code, and preventing its installation into the kernel proper. Our kernel memory enforcer works by marking the code section, sys call table, and other sensitive portions of the kernel as read-only through the

Claim Language	Exemplary Disclosures
	VMM. If a malicious program, such as a kernel back door tries to modify these sections of memory, the VM will be halted and the kernel memory protection enforcer notified.” 10 “INSPECTION COMMANDS are used to directly examine VM state such as memory and register contents, and I/O devices’ flags.” 6 “The OS interface library also facilitates queries at the level of kernel code, similar to the queries that one might give to gdb like “show me the contents of virtual memory from x to y in the context of the login process” or “display the contents of task structure for the process with PID 231.”... In order to support VMI, we added hooks to VMware to allow inspection of memory, registers, and device state. We also added hooks to allow interposition on certain events, such as interrupts and updates to device and memory state.” 7 Furthermore, this claim element is obvious in light of Garfinkel 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.
[4] The method of claim 1, further comprising: managing pestware-related events after the computer becomes configured to run native applications	Garfinkel discloses this limitation. For example: <i>See Claims [1A], [1C], [1D], [1G].</i> Furthermore, this claim element is obvious in light of Garfinkel 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
[5] The method of claim 1, wherein the monitoring includes monitoring the boot sequence while boot drivers are initiated.	Garfinkel discloses this limitation. For example: <i>See Claims [1C], [1E].</i> “[b]eing able to directly inspect the virtual machine makes it particularly difficult to evade a VMI IDS <i>since there is no state in the monitored system that the IDS cannot see.</i>” 4 Furthermore, this claim element is obvious in light of Garfinkel 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.
[6P] A system for managing pestware on a computer comprising:	To the extent the preamble may be construed to be limiting, Garfinkel discloses this limitation. For example: <i>See Claim [1P].</i> Furthermore, this claim element is obvious in light of Garfinkel 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;	Garfinkel discloses this limitation. For example: “A common way to virtualize the CPU is to run the VMM in the most privileged mode of the processor.” 1 “Among these techniques is configuring the real machine so that virtual machines can safely and directly execute using the machine’s CPU and memory.” 3

Claim Language	Exemplary Disclosures
	Furthermore, this claim element is obvious in light of Garfinkel 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.
[6B] a memory comprising:	Garfinkel discloses this limitation. For example: <i>See Claim [6A].</i> Furthermore, this claim element is obvious in light of Garfinkel 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;	Garfinkel discloses this limitation. For example: <i>See Claim [1A].</i> Furthermore, this claim element is obvious in light of Garfinkel 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	Garfinkel discloses this limitation. For example:

Claim Language	Exemplary Disclosures
pestware-related events during 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, and after a kernel is loaded;	<i>See Claim [1B].</i> Furthermore, this claim element is obvious in light of Garfinkel 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.
[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;	Garfinkel discloses this limitation. For example: <i>See Claim [1C].</i> Furthermore, this claim element is obvious in light of Garfinkel 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	Garfinkel discloses this limitation. For example: <i>See Claim [1D].</i>

Claim Language	Exemplary Disclosures
monitoring, a record of events, the record of events including the pestware-related events;	Furthermore, this claim element is obvious in light of Garfinkel 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.
[6G] a fifth instruction set configured to analyze the record of events so as to identify the pestware-related events; and	Garfinkel discloses this limitation. For example: <i>See Claim [1E].</i> Furthermore, this claim element is obvious in light of Garfinkel 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	Garfinkel discloses this limitation. For example: <i>See Claims [1E], [1F].</i> Furthermore, this claim element is obvious in light of Garfinkel 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	Garfinkel discloses this limitation. For example:

Claim Language	Exemplary Disclosures
in a registry of the computer for pestware during the second period in the boot sequence.	<i>See</i> Claim [1G]. Furthermore, this claim element is obvious in light of Garfinkel 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.
[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	Garfinkel discloses this limitation. For example: <i>See</i> Claim [2A]. Furthermore, this claim element is obvious in light of Garfinkel 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.	Garfinkel discloses this limitation. For example: <i>See</i> Claim [2B]. Furthermore, this claim element is obvious in light of Garfinkel 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
[8] The system 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.	Garfinkel discloses this limitation. For example: <i>See Claim [3].</i> Furthermore, this claim element is obvious in light of Garfinkel 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.
[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.	Garfinkel discloses this limitation. For example: <i>See Claims [1P], [1A], [1C], [1G].</i> Furthermore, this claim element is obvious in light of Garfinkel 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	Garfinkel discloses this limitation. For example: <i>See Claims [1E], [5].</i> Furthermore, this claim element is obvious in light of Garfinkel 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.
[11P] A non-transitory tangible computer-readable storage medium comprising program instructions for:	To the extent the preamble may be construed to be limiting, Garfinkel discloses this limitation. For example: <i>See Claims [1P], [6P].</i> Furthermore, this claim element is obvious in light of Garfinkel 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;	Garfinkel discloses this limitation. For example: <i>See Claim [1A].</i> Furthermore, this claim element is obvious in light of Garfinkel 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	Garfinkel discloses this limitation. For example: <i>See Claim [1B].</i> Furthermore, this claim element is obvious in light of Garfinkel 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
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;	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.
[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;	Garfinkel discloses this limitation. For example: <i>See Claim [1C].</i> Furthermore, this claim element is obvious in light of Garfinkel 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;	Garfinkel discloses this limitation. For example: <i>See Claim [1D].</i> Furthermore, this claim element is obvious in light of Garfinkel 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
[11E] analyzing the record of events so as to identify the pestware-related events;	Garfinkel discloses this limitation. For example: <i>See Claim [1E].</i> Furthermore, this claim element is obvious in light of Garfinkel 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.
[11F] modifying the set of behavior rules so as to prevent the pestware related events; and	Garfinkel discloses this limitation. For example: <i>See Claim [1F].</i> Furthermore, this claim element is obvious in light of Garfinkel 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.	Garfinkel discloses this limitation. For example: <i>See Claim [1G].</i> Furthermore, this claim element is obvious in light of Garfinkel 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.
[12A] The non-transitory tangible computer-readable storage medium of claim 11 further comprising program instructions for launching, after an operating system is initiated, a pestware management engine; and	Garfinkel discloses this limitation. For example: <i>See Claim [2A].</i> Furthermore, this claim element is obvious in light of Garfinkel 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.
[12B] wherein the set of behavior rules includes behavior rules compiled by the pestware management engine.	Garfinkel discloses this limitation. For example: <i>See Claim [2B].</i> Furthermore, this claim element is obvious in light of Garfinkel 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,	Garfinkel discloses this limitation. For example: <i>See Claim [3].</i>

Claim Language	Exemplary Disclosures
wherein the record of events includes information selected from a group consisting of: process identification information, file identification information, and hook generation information.	Furthermore, this claim element is obvious in light of Garfinkel 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.
[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.	Garfinkel discloses this limitation. For example: <i>See Claim [4].</i> Furthermore, this claim element is obvious in light of Garfinkel 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	Garfinkel discloses this limitation. For example: <i>See Claims [1E], [5].</i> Furthermore, this claim element is obvious in light of Garfinkel 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
boot drivers are initiated.	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 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.	Garfinkel discloses this limitation. For example: <i>See Claim [1F].</i> “We used a modified version of VMware Workstation for Linux to provide us with a virtual machine monitor capable of running common x86-based operating systems. In order to support VMI, we added hooks to VMware to allow inspection of memory, registers, and device state. <i>We also added hooks to allow interposition on certain events, such as interrupts and updates to device and memory state.</i>”⁷ “As part of this virtualization process, the VMM shadows the page tables of the physical machine, allowing the monitor to enforce more restrictive protection of certain memory pages. An example of how this functionality can be applied is the copy-on-write page sharing of the Disco virtual machine monitor [5]. We used this mechanism to write protect pages and provide notification if the VM attempted to modify a protected page.”⁷ “Each policy module is an individual Python module (i.e. a single file) that leverages the policy framework. Policy modules can be run stand-alone or in concert with other policy modules.”⁸ “LIVEWIRE FRONT END: The front end code is responsible for bootstrapping the system, starting the OS interface library process, loading policy modules, and running policy modules in concert. Interfaces are provided for obtaining configuration information, reporting intrusions, and registering policy modules with a common controller.”⁸ “Our Signature Scanner (“sig”) was able to detect all of the attacks we provided signatures for.... Our Raw Socket Detector (“raw”) raw socket detector detected the presence of cd00r and a similar stealth backdoor we included in our test attack suite. Our Lie Detector (“lie”) modules proved especially effective against rootkits both at kernel and user level since they all provided functionality to hide processes. The User Program Integrity Checker (“int”) was able to detect the presence of t0rn and lrk5 based on their use of backdoored version of inetd and sshd. We also found it also effective in detecting backdoored versions of syslog. Our

Claim Language	Exemplary Disclosures																																																																								
	Memory Protection Enforcer (“mem”) was able to detect and prevent the installation of all of kernel backdoors. knark and adore were stopped by blocking their attempt to modify sys call table. SUCKIT stopped by blocking its attempt to modify the interrupt dispatch table. Our NIC access enforcer (“nic”) was trivially able to detect and prevent the packet sniffers in our test attack suite from operating, based on their reliance on running the NIC in promiscuous mode.” 10-11 <table><tr><th>Name</th><th>Description</th><th>nic</th><th>raw</th><th>sig</th><th>int</th><th>lis</th><th>mem</th></tr><tr><td>cdor</td><td>Stealth user level remote backdoor</td><td></td><td>D</td><td></td><td></td><td></td><td></td></tr><tr><td>ifm</td><td>Precompiled user level rootkit</td><td></td><td></td><td>D</td><td></td><td>D</td><td></td></tr><tr><td>Ramen</td><td>Linux Worm</td><td></td><td></td><td>D</td><td></td><td></td><td></td></tr><tr><td>lk3</td><td>Source based user level rootkit</td><td>P</td><td></td><td>D</td><td>D</td><td>D</td><td></td></tr><tr><td>knark-0.59</td><td>LKM based kernel backdoor/rootkit</td><td></td><td></td><td>D</td><td></td><td>D</td><td>P</td></tr><tr><td>adore-0.42</td><td>LKM based kernel backdoor/rootkit</td><td></td><td></td><td>D</td><td></td><td>D</td><td>P</td></tr><tr><td>dmiff 2.4</td><td>All-purpose packet sniffer for switched networks</td><td>P</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>SUCKIT</td><td>/dev/kmem patching based kernel backdoor</td><td></td><td></td><td>D</td><td></td><td>D</td><td>P</td></tr></table> Table 1. Results of Livewire policy modules against common attacks. Within the grid, “P” designates a prevented attack and “D” a detected attack. “There are many sensitive mutable kernel data structures that we do not yet protect that could present an avenue for attack. We have simply tried to “raise the bar,” and prevent the most obvious of cases of this class of attack Finding better methods for identifying and enforcing the static and dynamic invariants that a VMI IDS relies upon seems an important area for further study.” 12 “MONITOR COMMANDS are used to sense when certainmachine events occur and request notification through an event delivery mechanism. For example, it is possible for a VMI to get notified when a certain range of memory changes, a privileged register changes, or a device state change occurs (e.g. Ethernet interface address is changed).” 6 “At the heart of any intrusion detection system is the policy engine. This component interprets system state and events from the VMM interface and OS interface library, and decides whether not the system has been compromised.” 7	Name	Description	nic	raw	sig	int	lis	mem	cdor	Stealth user level remote backdoor		D					ifm	Precompiled user level rootkit			D		D		Ramen	Linux Worm			D				lk3	Source based user level rootkit	P		D	D	D		knark-0.59	LKM based kernel backdoor/rootkit			D		D	P	adore-0.42	LKM based kernel backdoor/rootkit			D		D	P	dmiff 2.4	All-purpose packet sniffer for switched networks	P						SUCKIT	/dev/kmem patching based kernel backdoor			D		D	P
Name	Description	nic	raw	sig	int	lis	mem																																																																		
cdor	Stealth user level remote backdoor		D																																																																						
ifm	Precompiled user level rootkit			D		D																																																																			
Ramen	Linux Worm			D																																																																					
lk3	Source based user level rootkit	P		D	D	D																																																																			
knark-0.59	LKM based kernel backdoor/rootkit			D		D	P																																																																		
adore-0.42	LKM based kernel backdoor/rootkit			D		D	P																																																																		
dmiff 2.4	All-purpose packet sniffer for switched networks	P																																																																							
SUCKIT	/dev/kmem patching based kernel backdoor			D		D	P																																																																		

Claim Language	Exemplary Disclosures
	“LIVEWIRE FRONT END: The front end code is responsible for bootstrapping the system, starting the OS interface library process, policy modules, and running policy modules in concert. Interfaces are provided for obtaining configuration information, reporting intrusions, and registering policy modules with a common controller.” 8 “ADMINISTRATIVE COMMANDS allow the VMI IDS to control the execution of a VM. This interface allows the VMI IDS to suspend a VM’s execution, resume a suspended VM, checkpoint the VM, and reboot the VM. These commands are primarily useful for bootstrapping the system and for automating response to a compromise. A VMI IDS is only given administrative control over the VM that it is monitoring.” 6 Furthermore, this claim element is obvious in light of Garfinkel 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.
[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.	Garfinkel discloses this limitation. For example: <i>See Claims [1F], [1G].</i> “There are many sensitive mutable kernel data structures that we do not yet protect that could present an avenue for attack. We have simply tried to “raise the bar,” and prevent the most obvious of cases of this class of attack Finding better methods for identifying and enforcing the static and dynamic invariants that a VMI IDS relies upon seems an important area for further study.” 12 Furthermore, this claim element is obvious in light of Garfinkel 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.
[18] The method of claim 1, wherein the pestware-related events include events associated with at least one of loading pestware, executing pestware, and one or more behaviors of pestware.	Garfinkel discloses this limitation. For example: “Our Memory Protection Enforcer (“mem”) was able to <i>detect and prevent the installation</i> of all of our kernel backdoors. knark and adore were stopped by blocking their attempt to modify sys call table. SUCKIT was <i>stopped by blocking its attempt to modify</i> the interrupt dispatch table.” 10 “The NIC Access Enforcer prevents the Ethernet device entering promiscuous mode, or being configured with a MAC address which has not been pre-specified. Using this module <i>we can prevent variety of common misuses of the NIC to be detected and prevented.</i>” 10 Furthermore, this claim element is obvious in light of Garfinkel 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.
[19] The system of claim 6, wherein the pestware-related events include events associated with at least one of loading pestware, executing pestware, and one or more behaviors of pestware.	Garfinkel discloses this limitation. For example: <i>See Claim [18].</i> Furthermore, this claim element is obvious in light of Garfinkel 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
[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.	Garfinkel discloses this limitation. For example: <i>See</i> Claim [18]. Furthermore, this claim element is obvious in light of Garfinkel 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.