

EX 1010



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
60/484,085	07/01/2003	Brett M. Oliphant	

112117
Thomas D. Fortenberry, Attorney at Law
P.O. Box 2099
Woodville, TX 75979

CONFIRMATION NO. 1581
POA ACCEPTANCE LETTER



Date Mailed: 10/02/2014

NOTICE OF ACCEPTANCE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 10/01/2014.

The Power of Attorney in this application is accepted. Correspondence in this application will be mailed to the above address as provided by 37 CFR 1.33.

/tkim/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

POWER OF ATTORNEY TO PROSECUTE APPLICATIONS BEFORE THE USPTO

I hereby revoke all previous powers of attorney given in the application identified in the attached statement under 37 CFR 3.73(b).

I hereby appoint:



Practitioners associated with the Customer Number:

112117

OR



Practitioner(s) named below (if more than ten patent practitioners are to be named, then a customer number must be used):

Name	Registration Number	Name	Registration Number

as attorney(s) or agent(s) to represent the undersigned before the United States Patent and Trademark Office (USPTO) in connection with any and all patent applications assigned only to the undersigned according to the USPTO assignment records or assignment documents attached to this form in accordance with 37 CFR 3.73(b).

Please change the correspondence address for the application identified in the attached statement under 37 CFR 3.73(b) to:



The address associated with Customer Number:

112117

OR

☐ Firm or Individual Name			
Address			
City	State	Zip	
Country			
Telephone	Email		

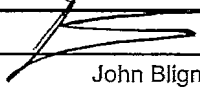
Assignee Name and Address:

SecurityProfiling, LLC
1919'S Shiloh Rd.
Garland, TX 75042

A copy of this form, together with a statement under 37 CFR 3.73(b) (Form PTO/SB/96 or equivalent) is required to be filed in each application in which this form is used. The statement under 37 CFR 3.73(b) may be completed by one of the practitioners appointed in this form if the appointed practitioner is authorized to act on behalf of the assignee, and must identify the application in which this Power of Attorney is to be filed.

SIGNATURE of Assignee of Record

The individual whose signature and title is supplied below is authorized to act on behalf of the assignee

Signature		Date	9.24.14
Name	John Blignaut	Telephone	
Title	Member - SecurityProfiling, LLC		

This collection of information is required by 37 CFR 1.31, 1.32 and 1.33. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 3 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

STATEMENT UNDER 37 CFR 3.73(b)

Applicant/Patent Owner: SecurityProfiling, Inc.

Application No./Patent No.: 60/484,085

Filed/Issue Date: 07-01-2003

Titled: Security management systems and methods

SecurityProfiling, LLC

, a Limited Liability Company

(Name of Assignee)

(Type of Assignee, e.g., corporation, partnership, university, government agency, etc.)

states that it is:

1. ☒ the assignee of the entire right, title, and interest in;
2. ☐ an assignee of less than the entire right, title, and interest in
(The extent (by percentage) of its ownership interest is _____ %); or
3. ☐ the assignee of an undivided interest in the entirety of (a complete assignment from one of the joint inventors was made)

the patent application/patent identified above, by virtue of either:

- A. ☐ An assignment from the inventor(s) of the patent application/patent identified above. The assignment was recorded in the United States Patent and Trademark Office at Reel _____, Frame _____, or for which a copy therefore is attached.

OR

- B. ☒ A chain of title from the inventor(s), of the patent application/patent identified above, to the current assignee as follows:

1. From: Brett M. Oliphant To: SecurityProfiling, Inc.

The document was recorded in the United States Patent and Trademark Office at
Reel 015544, Frame 0421, or for which a copy thereof is attached.

2. From: SecurityProfiling, Inc. To: SecurityProfiling, LLC

The document was recorded in the United States Patent and Trademark Office at
Reel _____, Frame _____, or for which a copy thereof is attached.

3. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at
Reel _____, Frame _____, or for which a copy thereof is attached.

☐ Additional documents in the chain of title are listed on a supplemental sheet(s).

- ☒ As required by 37 CFR 3.73(b)(1)(i), the documentary evidence of the chain of title from the original owner to the assignee was, or concurrently is being, submitted for recordation pursuant to 37 CFR 3.11.

[NOTE: A separate copy (i.e., a true copy of the original assignment document(s)) must be submitted to Assignment Division in accordance with 37 CFR Part 3, to record the assignment in the records of the USPTO. See MPEP 302.08]

The undersigned (whose title is supplied below) is authorized to act on behalf of the assignee.

/Thomas D. Fortenberry/

09-28-2014

Signature

Date

Thomas D. Fortenberry

Patent Attorney No. 56537

Printed or Typed Name

Title

This collection of information is required by 37 CFR 3.73(b). The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

FORTINET EX1010 Page 3

Privacy Act Statement

The **Privacy Act of 1974 (P.L. 93-579)** requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (*i.e.*, GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

EXHIBIT 1-ASSIGNMENT

WHEREAS, SecurityProfiling, Inc., having a place of business in Lafayette, IN (hereinafter "ASSIGNOR") is owner of:

USPN 8,266,699 filed on 07-01-2004
USAN 60/484,085 filed on 07-01-2003
USAN 10/882,788 filed on 07-01-2004
USAN 10/882,853 filed on 07-01-2004
USAN 10/882,852 filed on 07-01-2004
USAN 14/138,014 filed on 12-21-2013
USAN 10/883,376 filed on 07-01-2004
USAN 10/884,329 filed on 07-01-2004

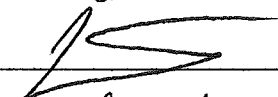
("Patent Asset(s)"),

WHEREAS, SecurityProfiling, LLC a Texas Limited Liability Company having a place of business in Texas (hereinafter "ASSIGNEE") desires to acquire ASSIGNOR's entire right, title, and interest in and to the Patent Asset(s);

NOW, THEREFORE, for good and valuable consideration, the receipt of which is hereby acknowledged, ASSIGNOR hereby acknowledges that it has sold, assigned, and transferred, and by these presents does hereby sell, assign, and transfer, unto ASSIGNEE, its successors, legal representatives, and assigns, the entire, irrevocable, and unconditional right, title, and interest of ASSIGNOR in, to, and under the Patent Asset(s), and the inventions disclosed in the Patent Asset(s) (regardless of whether claimed) including but not limited to (a) all rights of ASSIGNOR in any and all priority patent application(s), and all foreign and domestic patents that may issue from the Patent Asset(s) and the aforementioned priority patent application(s), including reexaminations, reissues, renewals, continuations, continuations-in part, divisionals, or extensions thereof that have been or may hereafter be filed, and (b) the right to sue for and collect damages for past, present, and future infringements of the Patent Asset(s).

IN TESTIMONY WHEREOF, I hereunto set my hand and seal this 23th day of September, 2014.

SecurityProfiling, Inc.

By: 

Name: John Blignaut

Title: CEO

Electronic Acknowledgement Receipt

EFS ID:	20294296
Application Number:	60484085
International Application Number:	
Confirmation Number:	1581
Title of Invention:	Security management systems and methods
First Named Inventor/Applicant Name:	Brett M. Oliphant
Correspondence Address:	Matthew R. Schantz Woodard, Emhardt, Moriarty, McNett & Henry LLP Bank One Center/Tower, Suite 3700 111 Monument Circle Indianapolis IN 46204-5137 US 3176343456 -
Filer:	THOMAS DONALD FORTENBERRY
Filer Authorized By:	
Attorney Docket Number:	
Receipt Date:	01-OCT-2014
Filing Date:	01-JUL-2003
Time Stamp:	00:05:39
Application Type:	Provisional

Payment information:

Submitted with Payment	no
------------------------	----

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Power of Attorney	POA_Pre-2012.pdf	66527 ea277b17db0ae0ecc4c377fd499f55c65dd86346	no	1
Warnings:					
Information:					
2	Assignee showing of ownership per 37 CFR 3.73.	02_3-73b.pdf	174021 3c03e845ab2bbb4b4db4895c3a82990c6712fce1	no	3
Warnings:					
Information:					
Total Files Size (in bytes):			240548		
This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503. <u>New Applications Under 35 U.S.C. 111</u> If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application. <u>National Stage of an International Application under 35 U.S.C. 371</u> If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course. <u>New International Application Filed with the USPTO as a Receiving Office</u> If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.					

SECURITY MANAGEMENT SYSTEMS AND METHODS

FIELD OF THE INVENTION(S)

The present invention(s) relates to computer security, and more particularly to systems and methods for managing and maintaining the security of networked computers.

DISCUSSION

For the purposes of promoting an understanding of the principles of the invention(s), reference will now be made to certain embodiments and features, and specific language will be used to describe the same. It will nevertheless be understood that no limitation of the scope of the invention(s) is thereby intended, such alterations and further modifications in and to the illustrated devices and methods, and such further applications of the principles of the invention(s) as illustrated therein being contemplated as would normally occur to one skilled in the art to which the invention(s) relate.

While the invention(s) have been illustrated and described in detail in the description below, the same is to be considered as illustrative and not restrictive in character, it being understood that only preferred embodiment(s) have been shown and described and that all changes and modifications that come within the spirit of the invention(s) are desired to be protected.

All publications, prior applications, and other documents cited herein are hereby incorporated by reference in their entirety as if each had been individually incorporated by reference and fully set forth herein.

Executive Summary

What is Anti-Vulnerability™ Technology?

Anti-Vulnerability technology is a platform and underlying back end that consists of a logic engine and vulnerability data warehouse. (Reference the diagram on page 8.) It will easily integrate with and enable network security products such as IDS, scanners, or firewalls to intelligently reference and share the same vulnerability data set, and independently provide complete vulnerability remediation (patching) functionalities such as that provided by the patch management and vulnerability remediation application, SysUpdate™. Thereby, the technology will improve system accuracy and efficiencies, minimize false positives and false negatives, and provide policy compliance and enforcement capabilities.

Market Requirements

The latest network compromises are combining qualities of viruses, worms, and denial of service attacks into new blended attacks. Additionally, blended attacks may now utilize metamorphic or polymorphic abilities to change their signatures to avoid detection.

To mitigate these new risks, organizations are now deploying a multi-tiered network defense strategy comprised of a variety of network security components layered at the perimeter and to internal network machines and devices. The security components are managed separately, and possibly their data is ported into a Security Information Management System (SIMS) correlation engine.

However, network security components and correlation engines rely on signatures or anomalies, producing an ever-increasing quantity of data, including false positive, benign, and erroneous events. They also lack the intelligence to identify, reference, or remediate the vulnerabilities targeted by the attack. Furthermore, the time and cost to acquire, manage, and maintain these numerous network security components is overwhelming the end user. Therefore, more sophisticated technology is needed to resolve these performance and management issues.

Anti-Vulnerability technology addresses these requirements by making each security component smarter and provides managers with remote remediation options. Thereby, improving system performance and streamlining management processes.

Therefore, a more sophisticated, intelligent technology is needed to resolve these issues.

Solution

SecurityProfiling now offers an intelligent integration platform and SDK — the first technology to meet market requirements.

Functional Description

1. **Enhances security products and provides system intelligence.** The same vulnerability data warehouse information is shared with the products so that more intelligent actions may then be taken to mitigate complex threats.
2. **Cross-references the threat's identifier with the target's configuration.** The threat's CVE ID, or other identifier, and the Destination IP address are fed into the logic engine where it cross-references the threat with the machine's existing OS, application, and patch configuration profile. An answer is returned accurately indicating if the machine is vulnerable.

-
3. **Identifies network vulnerabilities.** The platform will independently identify which network machines and devices have vulnerabilities by querying a client agent, or a device's OS or firmware directly.
 4. **Proactively remediates the vulnerability.** Once a vulnerability is identified, a remediation function is provided that may remotely and automatically deploy the appropriate update to each machine or device, install it, and document it with change tracking and audit trail reports.

Anti-Vulnerability™ Technology Applications

Examples of Anti-Vulnerability applications that may be integrated into network security products for enterprise and small office home office (SOHO) networks when integrated with Anti-Vulnerability technology via the SDK:

1. **Update or Patch Deployment** – Remotely remediates network vulnerabilities and provides policy compliance and enforcement capabilities. See *SecurityProfiling's* SysUpdate white paper and trial download online. And, for the consumer market, it will additionally support non-security related updates such as application version, performance, and bug-fix upgrades.
2. **IDS** – Accurately determines if the targeted machine is actually vulnerable to an attack; virtually eliminates false positives and false negatives; and adds vulnerability remediation and policy compliance and enforcement functionalities. (See *SecurityProfiling's* Intelligent IDS application.)
3. **Scanner** – Virtually eliminates false positive and false negatives, and adds vulnerability remediation and policy compliance and enforcement functionalities. (See *SecurityProfiling's* website, Workbench menu, for projects in development.)
4. **Firewall** – Virtually eliminates false positive and false negatives, preventing the security system from denying valid traffic to the organization – self-inflicted denial of service attacks - and adds vulnerability remediation, and policy compliance and enforcement functionalities.
5. **Vulnerability Management** – Same as 1., above.
6. **Multiple Products: IDS, Scanner, and Firewall** – Each may communicate with the same back-end data warehouse, sharing the same vulnerability information, attaining intelligence and more accurate information about the target and its vulnerabilities, thereby more effectively mitigating complex threats.
7. **Change Management** – Will automatically create a change request in the system, specifying what update/patch is applicable to what system or groups of systems by vulnerability. After approval of the request, they will automatically deploy and install the update/patch. Finally, they will also verify correct operation after installation and automatically provide all necessary documentation and logging required.

Lastly, the Anti-Vulnerability platform may also provide product upgrades and signature updates to each of these various security products.

Components

The Anti-Vulnerability platform is comprised of the following software components: SDK, Client Agent, Logic Engine, and Data Warehouse stored on a dedicated OnSite Server. Network Operations Center (NOC) Servers periodically synchronize the customers' servers with the latest vulnerability and update data. Please reference the Platform Component Diagram on page 8.

The same client agent and OnSite Server support all security applications that are integrated onto the platform, thereby providing the added Anti-Vulnerability functionalities presented in the previous section.

System Requirements

There are two system requirements: TCP/IP connectivity, and supported network machines and devices. Platforms supported are:

- All Windows Operating Systems and their applications
- All UNIX variants
- Cisco routers and firewalls
- Toshiba network devices
- Netgear network devices
- Linksys network devices, including wireless access points

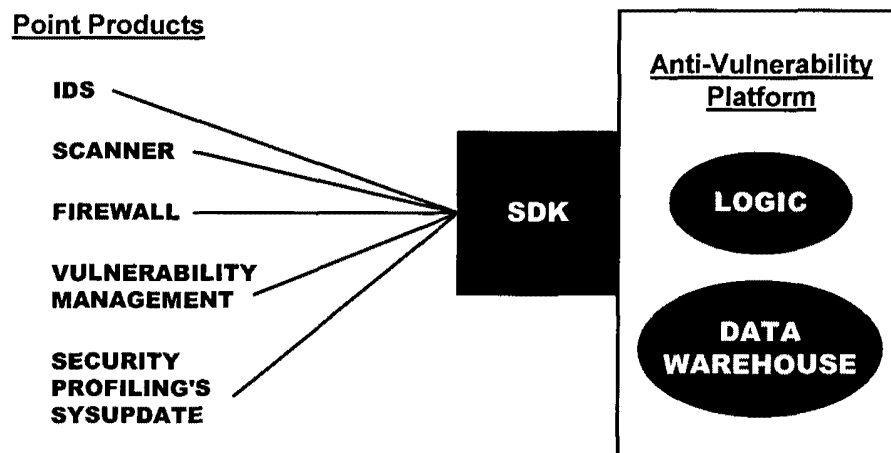
Platform support is scalable to any network size or architecture; consumer, small office home office (SOHO), enterprise, and distributed networks.

Support for additional machines and devices are in development and will be rolled out on an on-going basis.

What is Anti-Vulnerability™ Technology?

Anti-Vulnerability™ technology is a backend platform that consists of a logic engine and vulnerability data warehouse. It will easily integrate with and enable network security products such as IDS, scanners, or firewalls to intelligently reference and share the same vulnerability data set, and independently provide complete vulnerability remediation (patching) functionalities such as that provided by the patch management and vulnerability remediation application, SysUpdate™ and the Intelligent IDS. Thereby, the technology will improve system accuracy and efficiencies, minimize false positives and false negatives, and provide policy compliance and enforcement capabilities.

Figure I – Anti-Vulnerability Technology



For more detail, see the Platform Component Diagram on page 8.

Anti-Vulnerability technology will ideally meet market requirements, presented in the next section.

Market Requirements

Networks are probed at an ever-increasing frequency for vulnerabilities, which may be exploited with compromise attacks. To protect the network, administrators have traditionally set up perimeter defense strategies.

These strategies generally call for network managers to lock down core servers, and monitor/scan/filter all incoming and outgoing traffic at the network perimeter with several network security products such as antivirus and firewalls to identify and attempt to neutralize hackers and malicious code. In the past, these strategies worked well, however new threats are becoming more complex.

The latest malicious code may combine qualities of viruses, worms, and direct compromise attacks into new *blended* attacks. Virus payloads are becoming more complex and by using metamorphic or polymorphic abilities, viruses are able to change their signatures to avoid the fingerprint-type filtering that most applications employ.

To mitigate these new risks, organizations are now deploying a multi-tiered network defense strategy comprised of a variety of additional network security products layered at the perimeter and to internal network machines and devices. Such network security products include antivirus, firewall, scanners, and network and host based intrusion detection systems.

Each of these systems is based on specific signatures, rules, or anomalies of each attack and their variants, and do not identify and remediate the specific network vulnerabilities the attack is targeting. So each attack, and its variants, must be identified and analyzed, a signature prepared, then finally deployed to each point product on each customer's network. This process is uncoordinated among multiple disparate systems, and creates an ever-increasing number of signatures producing more and more attack alerts requiring immediate attention – many of which are erroneous. The security components are managed separately, and possibly their data is ported into a Security Information Management System (SIMS) correlation engine.

Additionally, to proactively remediate the vulnerabilities the malicious code is targeting, administrators must quickly and diligently update or “patch” each network machine and device, which is a time-consuming and costly process. Further compounding the burden on administrators, best practice and government compliance directives now require higher standards of network security and integrity to protect consumer privacy, and they must be documented with change tracking and audit trail reports.

Therefore, it is increasingly difficult and costly to effectively mitigate new threats and manage numerous systems — particularly in an environment of rising security standards and policy compliance requirements.

Thus, the multi-tiered network defense strategy is falling short primarily for four reasons:

1. The inability of multi-tiered network security products to communicate and share information necessary to mitigate blended threats, and minimize false positives and false negatives.
2. The vulnerabilities targeted by malicious code are not accurately identified nor proactively remediated.
3. The associated excessive aggregate cost of ownership of multiple systems.

-
4. The excessive administrative burden and cost of managing and maintaining multiple systems.

Hence, there is a need to integrate systems, share information intelligently to better defend against blended threats, reduce management and cost requirements, and automate vulnerability identification and remediation functionalities, as presented in the high-level requirements.

New Technology High-Level Requirements

A new technology is needed to meet market requirements:

1. Integrate network security products to share information
2. Provide system intelligence
3. Remediate network vulnerabilities

Solution

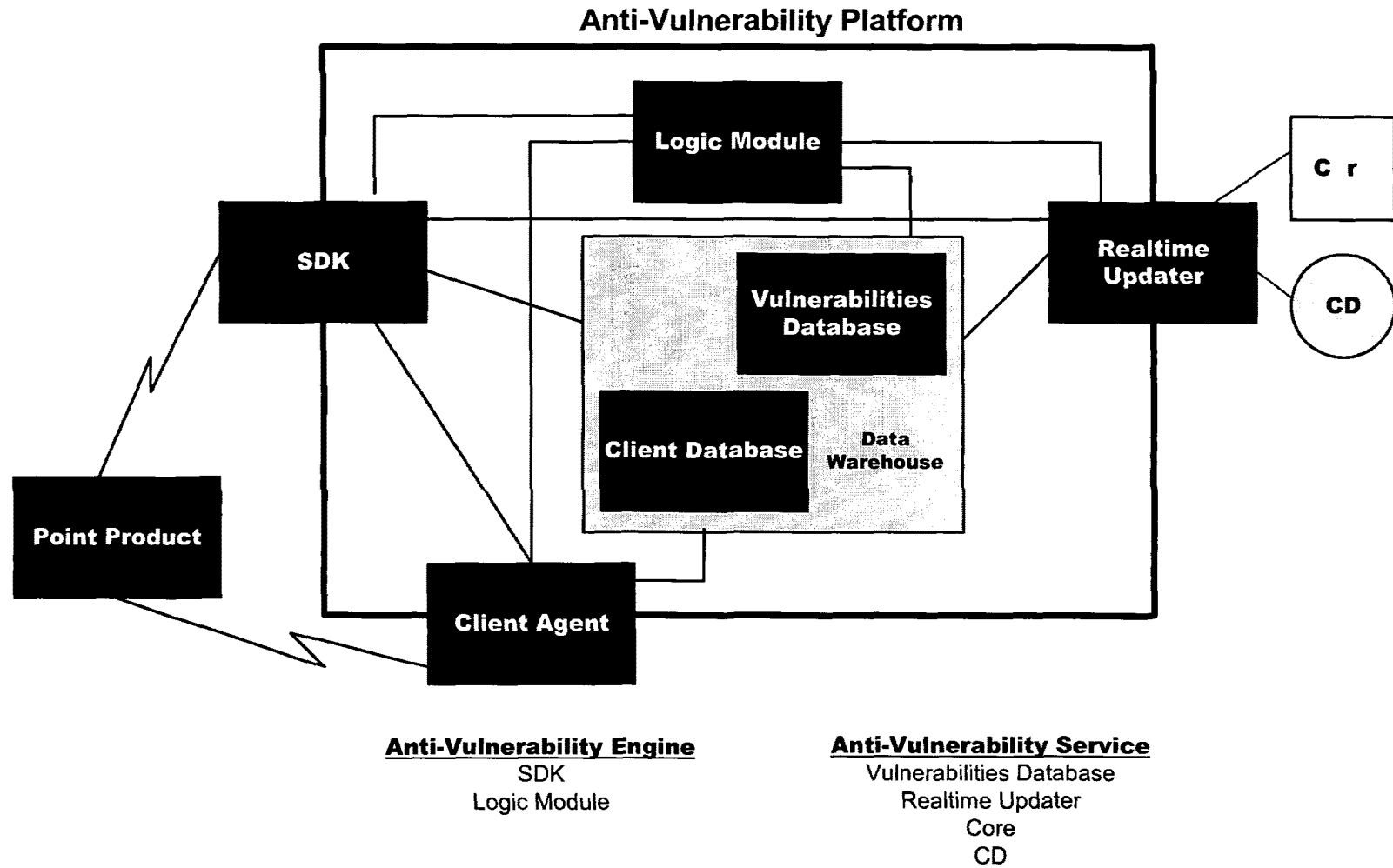
Anti-Vulnerability technology will integrate with and enable network security products to intelligently reference and share information from the same vulnerability data set, provide vulnerability identification and remediation functionalities, and efficiently meet policy compliance and enforcement requirements.

The Anti-Vulnerability platform is a complimentary network security technology. When integrated into the defense strategy, it adds intelligence that more accurately and efficiently mitigates blended threats and offloads the time-consuming functions that burden network administrators.

The platform enables network security products to share information via its proprietary **logic engine** to automatically cross-reference the threat identifier with the targeted machine's configuration to determine if it is actually vulnerable to that threat. Previously separate and uncoordinated processes are now more intelligent and automated, resulting in improved system accuracy and efficiency. Therefore the need to layer more and more point products and add a SIMS is reduced, in-turn decreasing the amount of data – particularly erroneous data – to manage. Subsequently, the costs to acquire, operate, and manage the additional multiple point products, and the need to implement a SIMS, are also reduced.

The platform will also remotely and proactively remediate vulnerabilities by first determining which updates are needed and compatible with each machine or device, taking into account the OS, applications, or firmware installed. Then, the updates may be deployed, installed, and validated. Thereby, policy compliance is effectively and efficiently enforced, and documented.

Figure 2 – SDK and Platform Component Diagram



Features & Benefits

Features:

Anti-Vulnerability technology fulfills market requirements noted in the previous section.

1. **Integrates network security products and provides system intelligence.** The same vulnerability data warehouse information is shared with all products so that more intelligent actions may then be taken to mitigate complex threats.
2. **Cross-references the threat's identifier with the target's configuration.** The threat's CVE ID, or other identifier, and the Destination IP address are fed into the logic engine where it cross-references the threat with the machine's existing OS, application, and patch configuration profile. An answer is returned accurately indicating if the machine is vulnerable. Thereby, minimizes erroneous, benign, and false positive data produced by each security product.
3. **Identifies network vulnerabilities.** The platform will independently identify which network machines and devices have vulnerabilities by querying a client agent, or a device's OS or firmware directly.
4. **Proactively remediates the vulnerability.** Once a vulnerability is identified, a remediation function is provided that may remotely and automatically deploy the appropriate update to each machine or device, install it, and document it with change tracking and audit trail reports.

Benefits:

Blended attacks are more effectively mitigated and the overwhelming management and maintenance burden on administrators to purchase, operate, and maintain multiple network security products is reduced, while increasing productivity, reducing costs, and more effectively and efficiently meeting policy compliance and enforcement requirements.

End Users

- Supplements existing technology to better defend against blended attacks
Intelligently accesses the vulnerability data warehouse, and remediates vulnerabilities.
- Offers policy compliance and enforcement functionality
Vulnerabilities may be automatically remediated, verified and documented, therefore enforcing compliance.
- No additional software or hardware implementation costs
Anti-Vulnerability functionalities may be integrated into existing product platforms.
- Reduces cost of ownership of multi network security products
Shared vulnerability data sets and added vulnerability remediation functionalities may reduce the number of network security products needed to adequately attain defense strategy requirements.

-
- Reduces management and maintenance costs

Increased accuracy of vulnerability identification, remediation and policy enforcement, and reduction of false positives, false negatives and denial of service (DoS), significantly reduces management time and costs.

- Manage more machines and devices on the network through one portal

Vulnerability remediation and policy compliance and enforcement may be integrated with existing security network security products, and their respective interfaces.

- Minimize end user education/ absorption costs

Anti-Vulnerability applications are transparently integrated, retaining pre-existing architecture, processes, and interfaces. Therefore, the end user experience remains the same, or improved with simplified or automated processes.

- User experience remains the same – additional time or costs to understand and execute new technologies are minimized

Anti-Vulnerability applications may be integrated transparently and seamlessly, and the pre-existing operational processes and user interfaces are virtually unchanged.

- Supports both the enterprise and SOHO networks

Data warehouse contains vulnerabilities and updates for many machines and devices that operate on both enterprise and SOHO networks. See the Supported Platforms subsection for more details.

Such applications are presented in the next section.

Anti-Vulnerability™ Technology Applications

Examples of Anti-Vulnerability applications that may be made available for enterprise and small office home office (SOHO) networks when integrated with Anti-Vulnerability technology via the SDK:

1. Update or Patch Deployment

SecurityProfiling's SysUpdate patch management and vulnerability remediation solution is the first Anti-Vulnerability technology application. A white paper and trial download are available online that demonstrate its Anti-Vulnerability functionalities and benefits.

Benefits: Anti-Vulnerability technology enables products to add accurate vulnerability identification, remediation, verification, and policy compliance functions. With Anti-Vulnerability technology, products will gain intelligence, accuracy and efficiency, eliminate false positives and false negatives, and ensure policy compliance, thereby saving the organization time and money. Such functionalities and benefits are available for both the enterprise and SOHO networks. And, for the consumer market, it will additionally support non-security related updates such as application version, performance, and bug-fix updates for widely distributed programs, which will optimize system performance and enhance the consumer experience.

2. IDS

IDS products attempt to identify malicious code by signatures at both the network and host client level. While they may be able to identify malicious code by CVE ID or other identifier, and targeted machines by IP address, but they generally do not have the intelligence to determine if any of the machines on the network are susceptible to that attack, or with finer granularity, if any machine has a specific vulnerability to that specific attack, or if the targeted vulnerability has already been patched. For example, if the malicious code has been written as a Windows based attack targeting a Windows vulnerability, is the Destination IP actually running Windows, or a UNIX variant? And, if Windows, is it vulnerable to the attack, or has it already been patched? IDS do not have the intelligence to answer these questions, and incident alerts are generated indiscriminately. Lastly, even if the targeted machine is vulnerable - it remains unremediated – an IDS does not have the capability to remediate it.

Benefits: With Anti-Vulnerability technology, a product architect can integrate functionalities listed in the previous section to enable the IDS to access the Anti-Vulnerability logic engine and platform. It will then have the intelligence to determine if any machine on the network is susceptible to the attack, remediate the vulnerability, mitigate the attack, and verify policy compliance. Now, if no machines were susceptible to the attack, it is identified as an event but not an incident, no further data or alert is generated, and a management response is not required. Integrated products will gain intelligence, accuracy, and efficiency, eliminate false positives and false negatives, and ensure policy compliance, thereby saving the organization time and money. Such functionalities and benefits are available for both the enterprise and SOHO networks.

3. Scanners

Vulnerability scanners assess each machine on the network for vulnerabilities, and create lists of potential vulnerabilities to the system managers. The lists commonly contain many false positives and false negatives, burdening the system and managers with inaccuracies.

Benefits: With Anti-Vulnerability technology, a product architect can integrate functionalities listed in the previous section to enable the scanner to access the Anti-Vulnerability logic engine and platform, and then have the intelligence to determine if the machine has actual vulnerabilities, remediate them, and verify policy compliance. Integrated products will gain intelligence, accuracy, and efficiency, eliminated false positives and false negatives, and ensured policy compliance, thereby saving the organization time and money.

4. Firewall

“Smart” firewalls are based in part on signatures and other similar functionalities as the IDS products described above.

Benefits: With Anti-Vulnerability technology, the firewall can determine whether an attack is valid or a false positive, thereby preventing the security system from denying valid traffic to the organization – self-inflicted DoS attacks. Such functionalities and benefits are available for both the enterprise and SOHO networks.

5. Vulnerability Management

Vulnerability Management products enable managers to set policy and identify potential network vulnerabilities. They typically do not accurately identify each vulnerability on each network machine and device, nor remediate each vulnerability, meeting policy compliance enforcement requirements.

Benefits: Anti-Vulnerability technology offers similar functionalities and benefits as the patch deployment products described above. In short, it enables products to add accurate vulnerability identification, remediation, verification, and policy compliance and enforcement. Such functionalities and benefits are available for both the enterprise and SOHO networks.

6. Multiple Products: IDS, Scanner, Firewall, and Vulnerability Management

Each product functions as the individual products noted above. Each may be deployed and operated on the network in a multi-tiered network defense strategy. They may be disparate system, and their signatures and update deployment schedules may vary. Therefore, the probability of mitigating a new and complex threat decreases, while management requirements and cost increases.

Benefits: With Anti-Vulnerability technology, they each may communicate with the same backend data warehouse, sharing the same vulnerability information, attaining intelligence and more accurate information about the target and its vulnerabilities, thereby more effectively mitigating complex threats. It will also enable the products to add accurate vulnerability identification, remediation, verification, and policy compliance and enforcement functionalities.

7. Change Management

Existing change management (CM) applications control the documentation and logging of change throughout the enterprise. These applications ensure that an organization maintains consistent records of what happened and when. Currently administrators must independently recognize that a security patch/update must be deployed to a computer or group of computers.

The user enters the request in the system, through the pipelined process of change management the request would be approved, the patch/update would be manually installed by a user, then documented in the change management software that the process has been completed. While CM software assists in documentation, very little if any assistance is provided to identify the patches/updates needed, nor verifying correct function after the update/patch is installed.

Benefits: With Anti-Vulnerability technology, change management integration will greatly streamline this process further reducing total cost of ownership, ease of use, and a higher standard of documentation. The products will then automatically create a change request in the system, specifying what update/patch is applicable to what system or groups of systems by vulnerability. After approval of the request, they will automatically deploy and install the update/patch. Finally, they will also verify correct operation after installation and automatically provide all necessary documentation and logging required.

Lastly, Anti-Vulnerability technology may also provide product upgrades and signature updates to each of these various security products - including all of the Anti-Vulnerability technology benefits such as ensuring compliance of signature versions, logging, reporting, and verification of installation.

Conclusion

Anti-Vulnerability technology is an intelligent platform that will supplement any existing defense strategy. Once integrated with the technology, security products may share the same vulnerability data set to improve automation and accuracy – increasing efficiencies and minimizing false positives and false negatives. It also enables remote identification, management, and remediation of network vulnerabilities, and provides update deployment, validation, and reporting capabilities. Thereby, the technology improves network security and integrity, mitigation of blended threats, while increasing productivity, reducing total cost ownership, and more effectively and efficiently attaining policy compliance and enforcement requirements.

Anti-Vulnerability SDK Function Call Summary

The SDK was designed to be a simple integration pathway to the Anti-Vulnerability platform backend. The SDK is documented and deliverable. See Appendix 1 for the function calls overview. The SDK is compatible with many different programming languages such as C, C++, VB, Java, and others.

The technology is sophisticated, yet all of the details of managing the communications are implemented in the library, requiring the developer to only learn a simple function calls as detailed in Appendix 1, and the SDK User's Guide. A product architect simply chooses the functionality desired, inserts the provided functionality SDK code into the product code. The SDK code will then query – via the logic engine – the Anti-Vulnerability data warehouse, and in turn will return an answer to the point product, as shown in Figure 2, on the next page. The architect can then show the results in any format desired in the product's interface. Full vulnerability remediation and policy compliance and enforcement functionalities may be integrated as well.

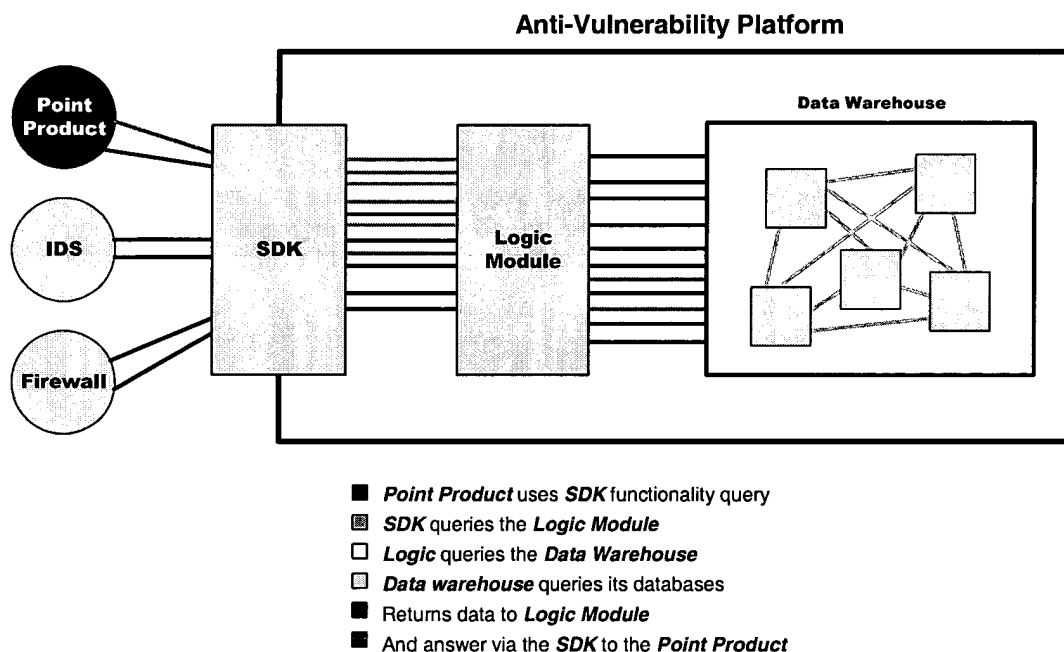


Figure 1 – SDK Function Call Diagram

Hence, the SDK serves as the architect's tool belt or building blocks needed to quickly and transparently access the Anti-Vulnerability backend and seamlessly integrate the desired functionality into the product without changing or complicating the end user experience.

Furthermore, the SDK enables system architects to use the Anti-Vulnerability platform to integrate multiple security products and functionalities. Products may then share the same vulnerability data set, improve system automation and accuracy – increasing efficiencies and minimizing false positives and false negatives, and coordinate policy compliance and enforcement functionalities.

Anti-Vulnerability functions that are accessible via the SDK. The functions are presented in groups, as follows:

1. **Session Management**
2. **Group Management Functions**
3. **Client Management Functions**
4. **Patch Management Functions**
5. **Polling Functions**
6. **User Management Functions**
7. **Miscellaneous Functions**
8. **Policy Compliance Functions**

1. Session Management Functions

- **OpenSession** – Parameters: HOST name or IP address, LOGIN and PASSWORD for host

Returns: new SESSION

Description: This function opens a new session to the OnSite Server

- **CloseSession** – Parameters: SESSION to close

Returns: None

Description: Closes a session that is no longer needed

- **IsSessionGood** – Parameters: SESSION to test

Returns: TRUE or FALSE

Description: returns TRUE if the session is still valid FALSE if it is not

- **SUGetLastError** – Parameters: None

Returns: Error code of last error in the thread

Description: Returns the error code of the last SDK error to occur in this thread

- **SUSetLastError** – Parameters: ERROR code to set

Returns: None

Description: Sets the error code for the thread to be returned by SUGetLastError

- **GetSDKVersion** – Parameters: None

Returns: a WORD containing the version of the SDK

Description: the MSB of the returned value is the major version number, the LSB is the minor version number

- **Ping** – Parameters: SESSION to use, VALUE to ping with

Returns: VALUE on success 0 on failure

Description: attempts to pass VALUE to the OnSite Server, which should pass it back.

2. Group Management Functions

- **CreateGroup** – Parameters: Group NAME and DESCRIPTION

Returns: GROUP that was just created or 0 on failure

Description: This function creates a new group with the provided name and description

- **GetGroups** – Parameters: None

Returns: A LIST of GROUPs.

Description: This function returns a list of all configured groups.

- **GetGroupsEx** – Parameters: None

Returns: A LIST of GROUPINFO structures

Description: This function returns a list of all configured groups' information

- **DeleteGroup** – Parameters: GROUP to be deleted.

Returns: Success or failure

Description: This function permanently removes a group.

-
- **EditGroup** – Parameters: GROUP to be edited, GROUPINFO to be applied
Returns: Success or failure
Description: This function changes a GROUP's information
 - **GetGroupPolicyOSCategories** – Parameters: GROUP to get information about
Returns: A LIST of OPERATINGSYSTEMCATEGORYs that have group policies configured
Description: This function retrieves a list of operating system categories that have group policies configured
 - **GetGroupPolicyOSSubCategories** – Parameters: GROUP to get information about, OPERATINGSYSTEMCATEGORY to get sub category information about
Returns: A LIST of OPERATINGSYSTEMSUBCATEGORYs
Description: This function retrieves a list of operating system sub-categories of OPERATINGSYSTEMCATEGORY that have policies for the GROUP
 - **GetGroupPolicyInfo** – Parameters: GROUPPOLICY to get information about, GROUPPOLICYINFO pointer to receive the information
Returns: Success or Failure
Description: This function fills the passed GROUPPOLICYINFO structure with the information for the GROUPPOLICY
 - **GetGroupPolicyInfoByOSCat** – Parameters: GROUP to get information about, OPERATINGSYSTEMCATEGORY to get the policy for, GROUPPOLICYINFO pointer to receive the information
Returns: Success or failure
Description: This function retrieves the information for a policy for the OPERATINGSYSTEMCATEGORY in the GROUP
 - **GetGroupPolicyInfoByOSSubCat** – Parameters: GROUP to get information about, OPERATINGSYSTEMSUBCATEGORY to get the policy for, GROUPPOLICYINFO pointer to receive the information
Returns: Success or failure
Description: This function retrieves the information for a policy for the OPERATINGSYSTEMSUBCATEGORY in the GROUP
 - **DeleteGroupPolicy** – Parameters: GROUPPOLICY to delete
Returns: Success or failure
Description: This function permanently deletes the GROUPPOLICY
 - **EditGroupPolicy** – Parameters: GROUPPOLICYINFO with new information
Returns: Success or failure
Description: This function updates the group policy (contained in GROUPPOLICYINFO) with the information in GROUPPOLICYINFO
 - **AddGroupPolicy** – Parameters: GROUP to add the policy to, GROUPPOLICYINFO containing the policy information
Returns: Success or Failure
Description: This function creates a new group policy and applies it to the GROUP
 - **GetClientsInGroup** – Parameters: GROUP to retrieve the clients from
Returns: LIST of CLIENTs belonging to the group
Description: This function retrieves a list of clients that are members of a specific group
 - **GetClientsInGroupEx** – Parameters: GROUP to retrieve the clients from
Returns: LIST of CLIENTINFOs for clients belonging to the group
Description: This function returns a list of CLIENTINFO structures containing information for the clients that belong to this group
 - **AddClientToGroup** – Parameters: CLIENT to add to a group, GROUP to add the client to.
Returns: Success or failure
Description: This function adds a client to a group
 - **RemoveClientFromGroup** – Parameters: CLIENT to remove from group, GROUP to remove the client from

Returns: Success or failure.

Description: This function removes a client from a group.

- **MoveClientToGroup** – Parameters: CLIENT to move, GROUP to move client to

Returns: Success or Failure

Description: This function moves CLIENT to GROUP

- **GetUngroupedClients** -- Parameters: None

Returns: LIST of ungrouped clients

Description: This function returns a LIST of the clients that do not belong to any group

- **GetUngroupedClientsEx** – Parameters: None

Returns: LIST of CLIENTINFO structures

Description: This function returns a LIST of CLIENTINFO structures for the clients that do not belong to any group

- **GetGroupInfo** – Parameters: GROUP to get information about, GROUPINFO pointer to receive the information

Returns: Success or failure

Description: This function retrieves the information for the GROUP

3. Client Management Functions

- **GetClientInstalledSoftware** – Parameters: CLIENT to get information about

Returns: LIST of SOFTWARE installed on the client

Description: This function retrieves a list of the software that has been detected on the client

- **GetClientInstalledSoftwareEx** – Parameters: CLIENT to get information about

Returns: LIST of SOFTWAREINFO for software installed on the client

Description: This function returns a LIST of SOFTWAREINFO structures describing the software detected on the client

- **GetClientInstalledPatches** – Parameters: CLIENT to get information from.

Returns: LIST of PATCHES corresponding to the installed patches on the client

Description: This function retrieves a list of patches that were verified as installed on the last polling

- **GetClientInstalledPatchesEx** – Parameters: CLIENT to get information from

Returns: LIST of PATCHINFO structures corresponding to the installed patches on the client

Description: This function retrieves a LIST of PATCHINFO structures for patches that were verified as installed on the last polling

- **GetClientPendingPatches** – Parameters: CLIENT to get information from.

Returns: LIST of PATCHES corresponding to the pending patches for the client

Description: This function returns a list of patches that the client needs (according to group policy) as of the last polling.

- **GetClientPendingPatchesEx** – Parameters: CLIENT to get information from

Returns: A LIST of PATCHINFO structures corresponding to the pending patches for the client

Description: This function returns a LIST of PATCHINFO structures for patches that the client needs (according to group policy) as of the last polling.

- **GetClientPendingServicePack** – Parameters: CLIENT to get information from

Returns: SERVICEPACK if a service pack is pending or 0 if not

Description: This function returns a SERVICEPACK if there is a service pack pending for the client (according to group policy) as of the last polling. If there is no service pack pending it returns 0

- **GetClientPendingSoftware** – Parameters: CLIENT to get information from

Returns: LIST of SOFTWARE that is pending

Description: This function returns a list of software that is pending for the client (according to group policy) as of the last polling

-
- **GetClientLogs** – Parameters: CLIENT to get information from
Returns: LIST of LOGs for the client.
Description: This function returns a list of logs for a particular client.
 - **GetClientLogsEx** – Parameters: CLIENT to get information from
Returns: LIST of LOGINFORM structures
Description: This function returns a list of LOGINFORM structures containing log entries for a particular client.
 - **DeleteClient** – Parameters: CLIENT to delete.
Returns: Success or failure.
Description: This function removes all information pertaining to this client from the SysUpdate database.
 - **GetClientOS** – Parameters: CLIENT to get information from
Returns: OPERATINGSYSTEM installed on client
Description: This function returns the OPERATINGSYSTEM installed on the client
 - **GetClientServicePack** – Parameters: CLIENT to get information from
Returns: SERVICEPACK installed on client or 0
Description: This function returns the service pack installed on the client or 0 if the client does not have a service pack installed, or if service packs are not applicable to the client's platform
 - **GetClientInfo** – Parameters: CLIENT to get information from CLIENTINFO pointer to receive the information
Returns: Success or failure
Description: This function fills the passed CLIENTINFO structure with information pertaining to the passed CLIENT
 - **GetClientFromIP** – Parameters: IP address of client
Returns: CLIENT that corresponds to the passed IP address
Description: This function maps IP addresses to clients

4. Patch Management Functions

- **GetPatchInfo** – Parameters: PATCH to get information about, PATCHINFO pointer to receive information
Returns: Success or failure
Description: This function fills the passed patch info structure with information about the passed patch
- **InstallPatchForClient** – Parameters: CLIENT to install the patch on, PATCH to install
Returns: Success or failure
Description: This function deploys the patch to the client. Installation proceeds asynchronously
- **InstallPatchForGroup** – Parameters: GROUP to install the patch on, PATCH to install
Returns: Success or failure
Description: This function deploys the patch to all clients in the group
- **InstallAllPendingForClient** – Parameters: CLIENT to install patches on
Returns: Success or failure
Description: This function deploys all pending patches for a client to the client
- **InstallAllPendingForGroup** – Parameters: GROUP to install patches on
Returns: Success or failure
Description: This function deploys all pending patches to all clients in a group
- **RejectPatchForClient** – Parameters: CLIENT to reject patch for, PATCH to reject
Returns: Success or failure
Description: This function rejects a patch for a client. This patch will not be queued down or installed

-
- **RejectPatchForGroup** – Parameters: GROUP to reject patch for, PATCH to reject
Returns: Success or failure
Description: This function rejects a patch for every client in a group. This patch will not be queued down or installed
 - **RequeuePatchForClient** – Parameters: CLIENT to requeue the patch for, PATCH to requeue
Returns: Success or failure
Description: This function requeues a previously rejected patch for a client
 - **RequeuePatchForGroup** – Parameters: GROUP to requeue the patch for, PATCH to requeue
Returns: Success or failure
Description: This function requeues a previously rejected patch for every client in a group
 - **RemovePatchFromClient** – Parameters: CLIENT to remove patch from, PATCH to remove
Returns: Success or failure
Description: This function will uninstall a patch from a client
 - **RemovePatchFromGroup** – Parameters: GROUP to remove patch from, PATCH to remove
Returns: Success or failure
Description: This function will uninstall a patch for every client in a group
 - **InstallServicePackForClient** – Parameters: CLIENT to install service pack on, SERVICEPACK to install on client
Returns: Success or failure
Description: This function will install a service pack on a client
 - **InstallServicePackForGroup** – Parameters: GROUP to install service pack on, SERVICEPACK to install
Returns: Success or failure
Description: This function will install a service pack on every client in a group
 - **InstallSoftwareForClient** – Parameters: CLIENT to install software update on, SOFTWARE to install
Returns: Success or failure
Description: This function will install a software update on a client
 - **InstallSoftwareForGroup** – Parameters: GROUP to install software update on, SOFTWARE to install
Returns: Success or failure
Description: This function will install a software update on every client in a group.
 - **GetCveIDsFromPatch** – Parameters: PATCH to get CveIDs from
Returns: A LIST of CVEIDs
Description: This function returns a list of CVEIDs the correspond to the PATCH

5. Polling Functions

- **PollClient** – Parameters: CLIENT to poll
Returns: Success or failure.
Description: This function causes a client to be polled by the OnSite Server immediately.
- **PollGroup** – Parameters: GROUP to poll
Returns: Success or failure.
Description: This function causes all the clients in a particular group to be polled immediately.
- **PollAll** – Parameters: NONE
Returns: Success or failure.
Description: This function causes all clients to be polled immediately.

6. User Management Functions

- **GetUsers** – Parameters: None

Returns: A LIST of USERS

Description: This function returns a list of all users for the system

- **GetUserInfo** – Parameters: USER to get information about, USERINFO pointer to receive the information

Returns: Success or failure

Description: This function populates the passed USERINFO structure with information about the passed USER

- **GetUserGroupPrivileges** – Parameters: USER to get information about

Returns: A LIST of USERGROUPPRIVILEGES

Description: This function returns a LIST of USERGROUPPRIVILEGES representing the privileges that a particular user has

- **GetUserGroupPrivilegesEx** – Parameters: USER to get information about

Returns: A LIST of USERGROUPPRIVILEGEINFO structures

Description: This function returns a LIST of USERGROUPPRIVILEGEINFO structures representing the privileges that a particular user has

- **GetUserGroupPrivilegeInfo** – Parameters: USERGROUPPRIVILEGE to get information about, USERGROUPPRIVILEGEINFO pointer to receive information

Returns: Success or failure

Description: This function populates the passed USERGROUPPRIVILEGEINFO structure with the details of the particular privilege

- **GetUserGroupPrivilegeInfoByGroup** – Parameters: USER to get information about, GROUP to get information for, USERGROUPPRIVILEGEINFO pointer to populate

Returns: Success or Failure

Description: This function populates the passed USERGROUPPRIVILEGEINFO structure with the detailed information regarding the permissions that the passed user has on the passed group

- **AddUserGroupPrivilege** – Parameters: USERGROUPPRIVILEGEINFO structure to add

Returns: Success or failure

Description: This function adds a USERGROUPPRIVILEGE

- **EditUserGroupPrivilege** – Parameters: USERGROUPPRIVILEGEINFO with new values

Returns: Success or failure

Description: This function edits the USERGROUPPRIVILEGEINFO structure passed in, assign it the new values

- **DeleteUserGroupPrivilege** – Parameters: USERGROUPPRIVILEGE to delete

Returns: Success or failure

Description: This function deletes the passed USERGROUPPRIVILEGE from the system

- **AddUser** – Parameters: USERNAME, PASSWORD, and USERTYPE

Returns: USER representing the user that was just created

Description: This function creates a new user with the provided username, password, and type, and returns a USER representing the new user or 0 in the case of an error

- **DeleteUser** – Parameters: USER to delete

Returns: Success or failure

Description: This function permanently deletes the USER from the system along with any privileges the user may have had

- **ChangeUserPassword** – Parameters: USER to change password for, new PASSWORD

Returns: Success or failure

Description: This function changes the password for a user

- **ChangeUserRole** – Parameters: USER to change role, ROLE to change to

Returns: Success or failure

Description: This Function changes a user's role

-
- **GetUserFromName** – Parameters: USERNAME
Returns: USER corresponding to the passed user name
Description: This function provides a mapping from user names to USERS

7. Miscellaneous Functions

- **CreateList** – Parameters: None
Returns: A new, empty, LIST
Description: This functions creates a new LIST that is initially empty
- **GetListItemEx** – Parameters: LIST to retrieve item from, INDEX of item to retrieve, VOID pointer to receive item
Returns: Success or failure
Description: This function populates the memory pointed to by VOID with the contents of the LIST item at INDEX
- **GetListSize** – Parameters: LIST to get the size of
Returns: the size of the LIST
Description: This function returns the number of elements currently in the list
- **AddItemToList** – pointer to LIST to add the item to, LONG item to add
Returns: None
Description: This function adds an “item” (such as a GROUP or a CLIENT) to a LIST
- **AddClientInfoExToList** – See AddItemToList
- **AddGroupInfoToList** – See AddItemToList
- **AddPatchInfoExToList** – See AddItemToList
- **AddSoftwareInfoToList** – See AddItemToList
- **AddSoftwareCategoryInfoToList** – See AddItemToList
- **AddUserGroupPrivilegeInfoExToList** – See AddItemToList
- **RemoveItemFromList** – Parameters: pointer to LIST to remove the item from, INDEX of item to remove
Returns: None
Description: This function removes the item at INDEX from the LIST
- **FreeList** – LIST to deallocate
Returns: None
Description: This function deallocates the resources used by LIST
- **GetSoftwareInfo** – Parameters: SOFTWARE to get information for, SOFTWAREINFO pointer to hold the information
Returns: Success or failure
Description: This function populates the SOFTWAREINFO structure pointed at with information regarding the SOFTWARE
- **GetLogInfo** – Parameters: LOG to get information about, LOGININFO pointer to contain the information
Returns: Success or failure
Description: This function populates the LOGININFO structure pointed at with information regarding the LOG
- **GetSoftwareCategories** – Parameters: None
Returns: A LIST of SOFTWARECATEGORYs
Description: This function returns a LIST of SOFTWARECATEGORYs
- **GetSoftwareCategoriesEx** – Parameters: None
Returns: A LIST of SOFTWARECATEGORYINFO structures
Description: This function returns LIST of SOFTWARECATEGORYINFO structures for every software category known on the system
- **GetSoftwareCategoriesForOperatingSystemCategory** – Parameters: an OPERATINGSYSTEMCATEGORY to get information for
Returns: A LIST of SOFTWARECATEGORYs
Description: This function returns a LIST of software categories for the passed operating system category.

-
- **GetSoftwareCategoriesForOperatingSystemCategoryEx** – As above only, returning the appropriate “info” structures in the list
 - **GetSoftwareCategoryInfo** – Parameters: SOFTWARECATEGORY to get information for, SOFTWARECATEGORYINFO pointer to contain the information
Returns: Success or failure
Description: This function fill the passed SOFTWARECATEGORYINFO structure with information about the passed SOFTWARECATEGORY
 - **GetOperatingSystemInfo** – Parameters: OPERATINGSYSTEM to get info about, OPERATINGSYSTEMINFO pointer to contain the data
Returns: Success or failure
Description: This function fills the OPERATINGSYSTEMINFO structure pointed to with information about the OPERATINGSYSTEM
 - **GetServicePackInfo** – As above for SERVICEPACK and SERVICEPACKINFO
 - **GetCveIDInfo** – Parameters: a CVEID to get information for, and a STRING to receive the information
Returns: Success or failure
Description: This function fills the STRING with the name of the CVEID (ie. CVE-2003-0344)
 - **MakeCveID** – Parameters: A CHAR pointer containing the name of a Cve ID
Returns: A CVEID
Description: This function maps character representations of Cve Ids to CVEIDs
 - **GetOperatingSystemCategories** – Parameters: None
Returns: A LIST of OPERATINGSYSTEMCATEGORYs
Description: This function returns a list of OPERATINGSYSTEMCATEGORYs available on the system
 - **GetOperatingSystemSubCategories** – Parameters: OPERATINGSYSTEMCATEGORY to get sub categories for
Returns: A LIST of OPERATINGSYSTEMSUBCATEGORYs
Description: This function returns a list of subcategories for the passed category
 - **GetOperatingSystemCategoryInfo** – Same as GetOperatingSystemInfo for OPERATINGSYSTEMCATEGORY and OPERATINGSYSTEMCATEGORYINFO
 - **GetOperatingSystemSubCategoryInfo** – Same as above for OPERATINGSYSTEMSUBCATEGORY and OPERATINGSYSTEMSUBCATEGORYINFO
 - **GetServicePacksForOperatingSystemSubCategory** – Parameters: OPERATINGSYSTEMSUBCATEGORY to get service packs for
Returns: A LIST of SERVICEPACKs
Description: this function retrieves a list of service packs that are valid for the operating system subcategory
 - **DoCoreUpdate** – Parameters: INT to indicate whether or not to force the update
Returns: Success or failure
Description: This function causes the OnSite server to do a core update. If the FORCE parameter is nonzero the update is forced
 - **QueryVulnerabilityByCveID** – Parameters: CLIENT to query, CVEID to check for
Returns: An error code indicating vulnerability
Description: This function returns an error code indicating whether or not the client is vulnerable to a CVEID
 - **GetDatabaseInfo** – Parameters: DATABASEINFO pointer to receive the information
Returns: Success or failure
Description: This function returns the information needed to connect to the OnSite Server’s database in the passed DATABASEINFO structure

8. Policy Compliance Functions

These functions are provided in a separate source tree (policy tree) for the Policy Compliance and Enforcement module. Function include: Password/system access policies, log configurations, audit settings, user/group privilege rights, general service settings, service permissions, registry values, registry permissions, file permissions. It can also perform deltas on and get information about: groups, users, services, and all the "Run" registry keys.

The function calls are being integrated into the live production tree and subject to change.

- **GetPolicyItemInfo** – As GetOperatingSystemInfo for POLICYITEM and POLICYITEMINFO
- **InstallPolicyItemForClient** – As InstallPatchForClient for POLICYITEM
- **InstallPolicyItemForGroup** – As InstallPatchForGroup for POLICYITEM
- **InstallAllPendingPolicyItemsForClient** – As InstallAllPendingPatchesForClient for POLICYITEM
- **InstallAllPendingPolicyItemsForGroup** – As InstallAllPendingPatchesForGroup for POLICYITEM
- **RemovePolicyItemFromClient** – As RemovePatchForClient for POLICYITEM
- **RemovePolicyItemFromGroup** – As RemovePatchFromGroup for POLICYITEM

INTELLIGENT IDS

Provides IDS intelligence, accuracy and remote patching

functions - Snort IDS data output has been integrated with the Anti-Vulnerability™ platform via the Anti-Vulnerability SDK. The platform's underlying backend consisting of a logic engine and vulnerability data warehouse provides the added functions and performance.

Market Requirements

Conventional IDS produce an ever-increasing quantity of alert data, including erroneous and false positive data. They also lack the intelligence to identify or remediate the vulnerabilities targeted by the attack. Furthermore, the cost of the acquiring, managing, and maintaining conventional IDS is overwhelming the end user.

Therefore, a more sophisticated, intelligent technology is needed to resolve these issues.

Solution

SecurityProfiling offers an intelligent IDS — the first technology to meet market requirements. The alert data output from a standard distribution of Snort is integrated with the Anti-Vulnerability platform via the Anti-Vulnerability SDK, which will add the following functions.

Functional Description

- 1. Cross-references the threat's identifier with the target's configuration.** The CVE ID, or other identifier, and the Destination IP address are fed into the logic engine where it cross-references the threat with the machine's configuration profile.
- 2. Virtually eliminates false positives and false negatives.** The backend accurately determines in real time if the targeted machine is susceptible to the attack. And, if the machine is not susceptible, it is filtered and reported back as an event and not an incident. No further data or alert is generated, and a management response is not required.
- 3. Remotely remediates the vulnerability.** When a machine is identified as vulnerable to an attack, an incident, a remediation function is provided to the administrator to remotely deploy the appropriate update to the machine or device, install it, verifies the files and hashes, and document it with change tracking and audit trail reports.

Unique Features:

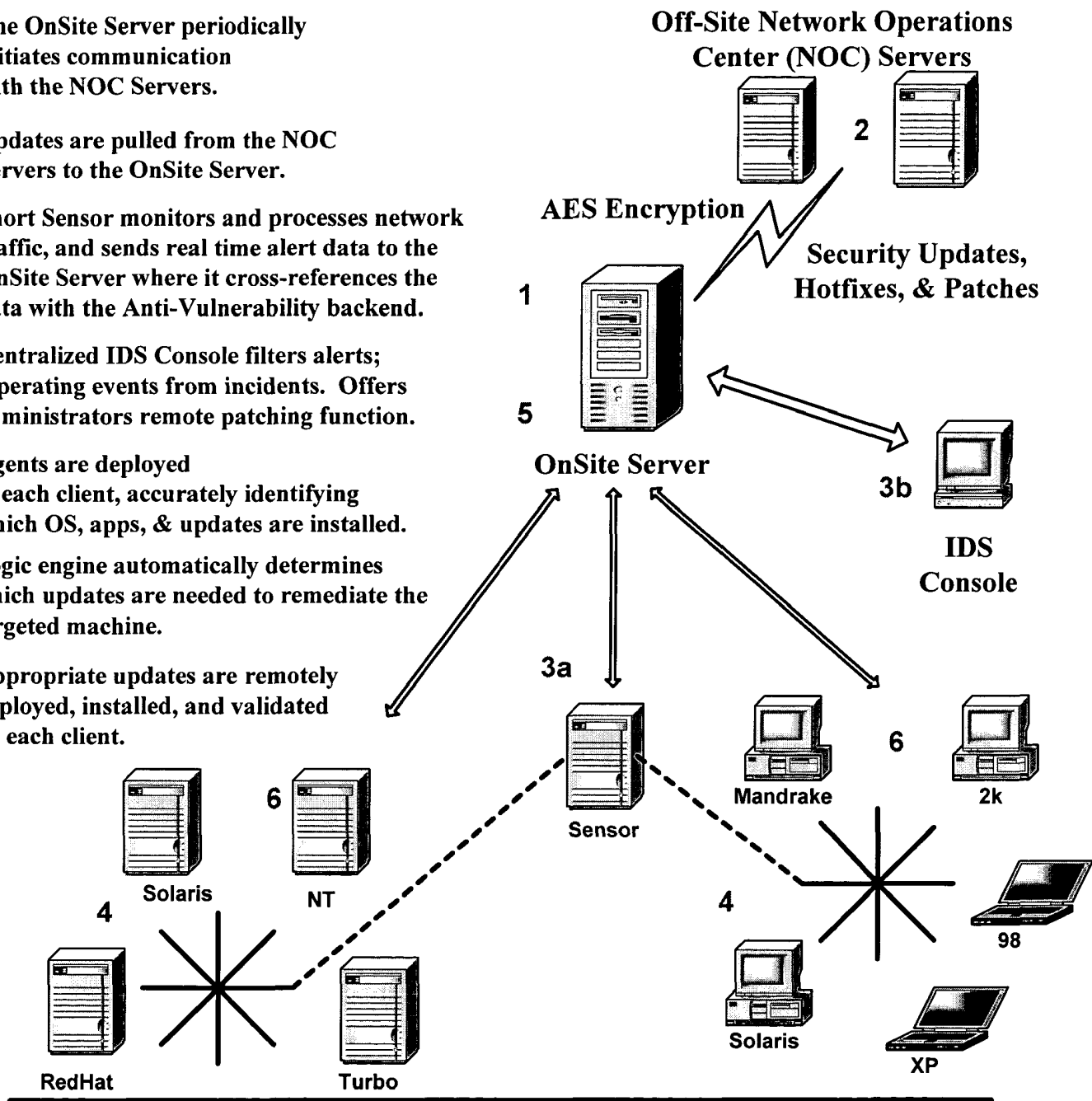
- Supports Windows and UNIX variants
- Determines if targeted machine is vulnerable to an attack in real time
- Filters out erroneous, benign and false positive alerts
- Remotely patches targeted vulnerabilities in one click
- Installs in minutes

Benefits

Erroneous, benign, and false positive data is filtered out, and incidents may be remotely remediated. Therefore, complex threats are more effectively and efficiently mitigated. Correspondingly, the management and maintenance burden on administrators is reduced, saving time and effort.

Intelligent IDS Configured Network Overview Diagram

- 1 The OnSite Server periodically initiates communication with the NOC Servers.
- 2 Updates are pulled from the NOC Servers to the OnSite Server.
- 3a Snort Sensor monitors and processes network traffic, and sends real time alert data to the OnSite Server where it cross-references the data with the Anti-Vulnerability backend.
- 3b Centralized IDS Console filters alerts; separating events from incidents. Offers administrators remote patching function.
- 4 Agents are deployed to each client, accurately identifying which OS, apps, & updates are installed.
- 5 Logic engine automatically determines which updates are needed to remediate the targeted machine.
- 6 Appropriate updates are remotely deployed, installed, and validated on each client.



SECURITYPROFILING

839 Main St. Suit 405 Lafayette, IN 47901-1403
 Tel: 888-645-3676 Fax: 765-420-9256
 E-Mail: info@securityprofiling.com
 Web: www.securityprofiling.com

A white paper and trial download are available online at www.securityprofiling.com



SYSUPDATE™

The Intelligent Patch Management and Vulnerability

Remediation Application - automatically updates computer OS and application vulnerabilities before they can be exploited by hackers and viruses.

Market Requirements

Networks are probed at an ever-increasing frequency for vulnerabilities, which may be exploited with directed compromise attacks. To protect the network, administrators must diligently update or “patch” server and workstation vulnerabilities, which is a time-consuming and costly process. Further compounding the burden on administrators, best practice and government compliance directives now require higher standards of network security to protect consumer privacy and proprietary data, which must be documented with change tracking and audit trail reports. Therefore, a new fully automated technology is needed to resolve these issues.

Solution

SecurityProfiling offers a new intelligent application called SysUpdate — the first technology to fully automate the updating and reporting processes, as described below.

Functional Description

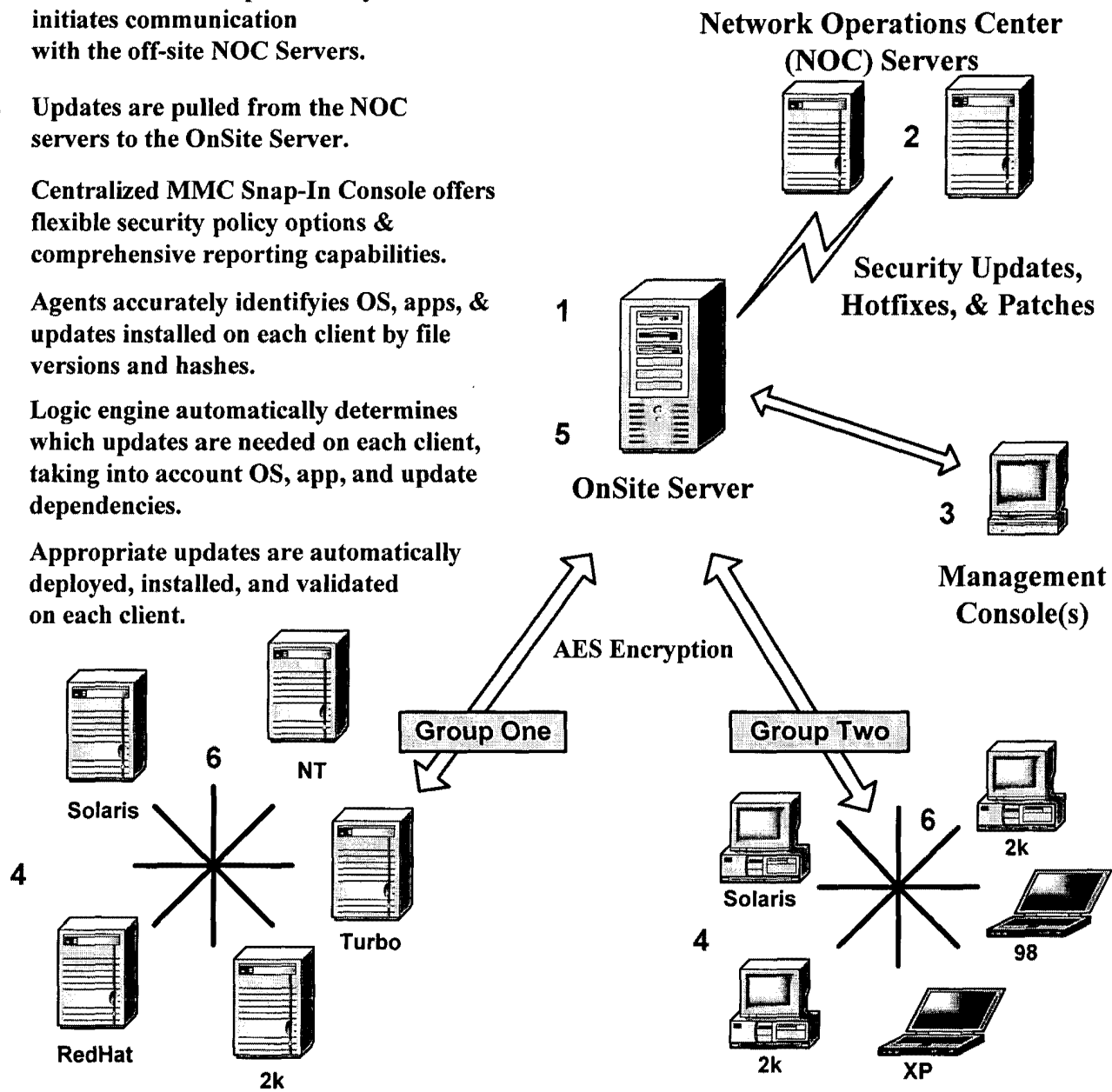
- 1. Automatically researches updates.** In collaboration with software development companies, SysUpdate receives updates and their respective compatibility and installation guidelines, which are thoroughly reviewed and tested in a lab for system compatibility and stability.
- 2. Automatically and securely downloads, and archives, all updates.** Once the updates and guidelines are thoroughly reviewed and tested, they are automatically downloaded to each customer’s OnSite Server, verified with MD5 hashes, and archived in a self-populating patch repository database.
- 3. Automatically determines the update and dependency requirements for all computers on a network, based on their individual operating systems, applications, and previously installed updates.** The system’s logic engine automatically matches the tested updates and their guidelines across each computer’s configuration profile.
- 4. Remotely, securely, and automatically deploys updates to each computer and device.** Updates may be tested on a test group, and then queued for release to each computer on the network, consistently in accordance with the policy determined by the network administrator.
- 5. Automatically verifies the updates are installed and running correctly.** Each computer’s client agent installs the updates, and verifies the files and hashes.
- 6. Automatically generates change tracking and audit trail reports.** Change tracking and audit trail reports may be selected and printed from the Management Console’s Crystal Reports modules. Additionally, application license audit reports may be generated, which lists software installed on each machine, by license key.

Unique Features:

- **Supports Windows, UNIX variants, and network devices**
- **Supports non-Microsoft applications**
- **Logic engine provides system intelligence**
- **Files verified with SHA1 or MD5 hashes**
- **MMC snap-in console with Crystal Reports**
- **AES encryption**

SysUpdate™ Configured Network Overview Diagram

- 1 The OnSite Server periodically initiates communication with the off-site NOC Servers.
- 2 Updates are pulled from the NOC servers to the OnSite Server.
- 3 Centralized MMC Snap-In Console offers flexible security policy options & comprehensive reporting capabilities.
- 4 Agents accurately identify OS, apps, & updates installed on each client by file versions and hashes.
- 5 Logic engine automatically determines which updates are needed on each client, taking into account OS, app, and update dependencies.
- 6 Appropriate updates are automatically deployed, installed, and validated on each client.



839 Main St. Suite 405 Lafayette, IN 47901-1403
 Tel: 888-645-3676 Fax: 765-420-9256
 E-Mail: inf@securityprofiling.com
 W b: www.securityprofiling.com

A white paper and trial download are available online at www.securityprofiling.com



LOGBOSS™

A secure computer log management application – with fully automated archiving, querying, and reporting options

Market Requirements

Best practice directives and government compliancy regulations now require administrators to archive log events over an extended period of time, and extensively document log and audit trail reports. With existing tools, these are time-consuming processes. Furthermore, existing tools transfer logs in clear text and are subject to malicious manipulation, and logs may be lost since few utilize guaranteed delivery protocols. Therefore, the integrity and security of the log data is not assured. In the event of an internal or external compromise, logs may be deleted, manipulated, or the network flooded with decoy log data.

Solution

LogBoss automatically and continuously transfers logs from each computer into a self-configuring, self-populating, maintenance-free database where they may be archived for years on end.

The transfer process utilizes AES encryption for authentication between the client and server, and a guarantee delivery protocol — ensuring no data is compromised or lost.

Flexible cross-correlation queries may be run from a single, remote viewer console, and the data is then automatically formatted into reports.

Cross-Correlation Query Type Descriptions

LogBoss offers the following three flexible cross-correlation queries, which provide increasing detail and breadth of filtering/sorting capabilities, respectively:

1. **General Queries** – This is the default query. It filters or sorts the logs of any one computer, by any one of the three log types.
2. **Specific Queries** – Detailed queries across all machines, and all log types.
3. **Advanced Queries** – Query across any combination of machines, by any combination of sources, events, and log types, over any period of time.

Automated Reports

Query data is automatically formatted into professional reports. Also, a separate reporting interface is available to generate general, machine, and user statistics. LogBoss quickly and easily fulfills best practice and government compliance requirements for log event archiving and reporting.

Unique Features:

- Continuously pulls all logs into a central database, which may be archived for years on end
- AES encryption and a guarantee delivery protocol ensure logs are not compromised or lost
- Queries may be across any computer, for any log type, over any time frame
- Automatically generates event log reports — ideal for documenting audit and compliance requirements
- Easy download installation

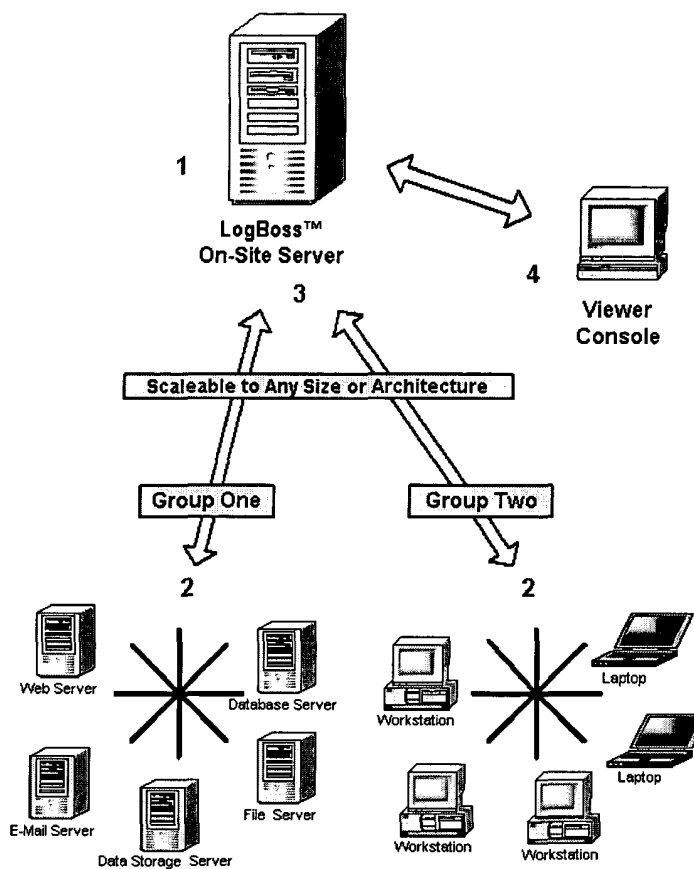
LogBoss™ Configured Network Overview Diagram

- 1 LogBoss server component automatically configures the application, and the database, and communicates with the client agents.
- 2 Client agents are deployed, which then identifies, manages, and continuously transfers all logs to the server.
- 3 All client log data will be automatically archived the self-populating database, for years on end.
- 4 From the centralized viewer, general, specific, or advanced cross-correlation queries may be run:

<u>Filter/ Sort By</u>	<u>General</u>	<u>Specific</u>	<u>Advanced</u>
# of Computers	Any One	Any One	Any
# of Log Types	Any One	All	Any
Period of Time	Last or All	Last or All	Any
# of Event	By ID Only	Any One	Any
Detail Fields*			

* Defined as Time/Date Written, UserID, ComputerID, EventID, Source, Type, & Category

Query and statistics data is automatically formatted into professional reports to easily document audit and compliance requirements.



839 Main St., Suite 405, Lafayette, IN 47901
Toll Free: 888.645.3676 ■ Phone: 765.420.7227 ■ Fax: 765.420.9256

A white paper and trial download are available online at www.securityprofiling.com



SYSUPDATE™ POLICY COMPLIANCE AND ENFORCEMENT MODULE

— The SysUpdate integrated module provides best practice security policy templates to proactively and remotely manage and enforce enterprise security compliance policies.

Market Requirements

Best practice and government compliance directives now require higher standards of network security to protect consumer privacy and proprietary data, which must be consistently enforced, and documented with change tracking and audit trail reports for compliance verification. Therefore, a fully automated technology is needed to resolve these issues.

Solution

SecurityProfiling offers a new intelligent application to manage and enforce security policies — the first technology to fully automate the policy configuration, enforcement, and reporting processes, as described below.

Functional Description

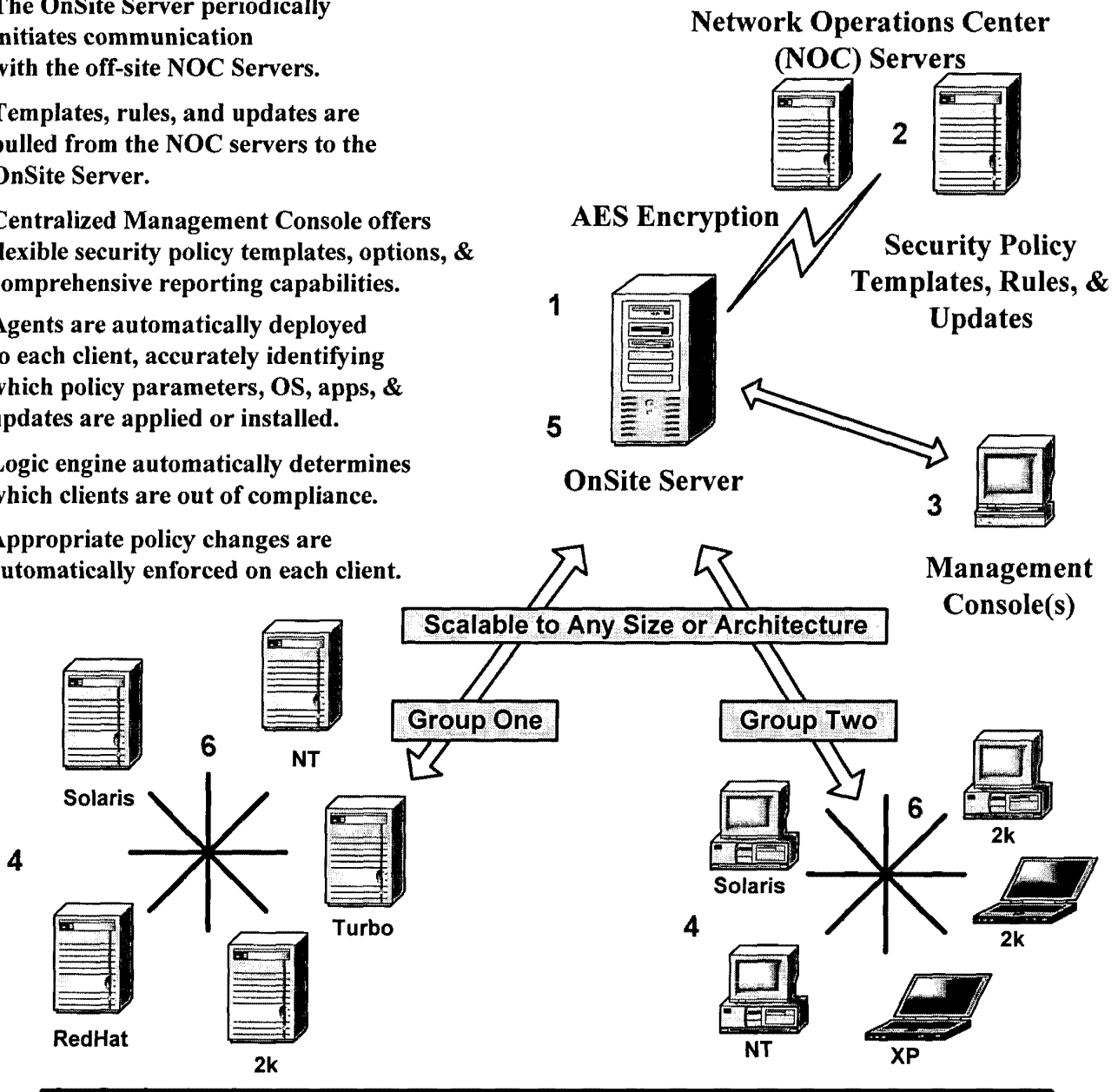
- 1. Centralized, Remote Management.**
The policy compliance and enforcement module is managed via the SysUpdate centralized management console - in MMC format, and automates the deployment, assessment, and enforcement of the policy chosen.
- 2. Standardized Policies.** Select from many standardized policy templates recommended by NSA, SANS, US Navy, and ISO 17799, or custom policy options to formulate policies that meet the individual needs of any organization.
- 3. Automatically determines out-of-compliance parameters for all computers and devices on a network.** Based on each client's individual policy parameter settings and configuration profile, the system's logic engine automatically reports back each client's out-of-compliance parameters.
- 4. Remotely, securely, and automatically enforces policy to each computer or device.** Policy changes may be automatically enforced to each computer or device individually, or by group, consistently in accordance with the policy selected by the network administrator. Policy configuration profiles are transmitted between client agent and server in an AES encrypted format for security and privacy. Policy options may be rolled back with one-click.
- 5. Automatically generates change tracking and audit trail reports.** Change tracking and audit trail reports may be selected and printed from the Management Console's Reporting Center.

Unique Features:

- Supports Windows, UNIX variants, and network devices
- Supports NSA, SANS, Navy, ISO 17799 & other standardized policies
- Logic engine provides system intelligence
- One-click policy rollback
- Automatically generates change tracking and audit trail reports

Policy Compliance and Enforcement Module Configured Network Overview Diagram

- 1 The OnSite Server periodically initiates communication with the off-site NOC Servers.
- 2 Templates, rules, and updates are pulled from the NOC servers to the OnSite Server.
- 3 Centralized Management Console offers flexible security policy templates, options, & comprehensive reporting capabilities.
- 4 Agents are automatically deployed to each client, accurately identifying which policy parameters, OS, apps, & updates are applied or installed.
- 5 Logic engine automatically determines which clients are out of compliance.
- 6 Appropriate policy changes are automatically enforced on each client.



839 Main St. Suite 405 Lafayette, IN 47901-1403
 T l: 888-645-3676 Fax: 765-420-9256
 E-Mail: info@securityprofiling.com
 W b: www.securityprofiling.com

A white paper and trial download are available online at www.securityprofiling.com

Provides intelligence, accuracy, real-time prevention, & remote patching functions – Accurately identifies and prevents malicious code from reaching their destination at the Inline IPS Sensor. Thereafter, a security officer may remotely patch the targeted vulnerability.

Conventional IDS/IPS produce an ever-increasing quantity of alert data, including erroneous and false positive data. They also lack the intelligence to identify or remediate the vulnerabilities targeted by the attack. Furthermore, the cost of the acquiring, managing, and maintaining conventional IPS is overwhelming the end user.

Solution

SecurityProfiling offers an intelligent IPS — the first technology to meet market requirements. The alert data output from a standard distribution of Snort is integrated with the Anti-Vulnerability platform via the Anti-Vulnerability SDK which accurately identifies attacks, and the attack is terminated at the Inline Sensor – as described in the following functions.

- 1. Cross-references the threat's identifier with the target's configuration.** The Inline Sensor monitors and processes traffic and sends alert data to the Anti-Vulnerability OnSite Server where its logic engine queries the backend in real-time to determine if the destination IP is vulnerable to the attack.

- Enhanced Flex Response.** If the destination IP is vulnerable to the attack, the Inline Sensor is commanded to immediately drop the exploit packets – preventing the attack.

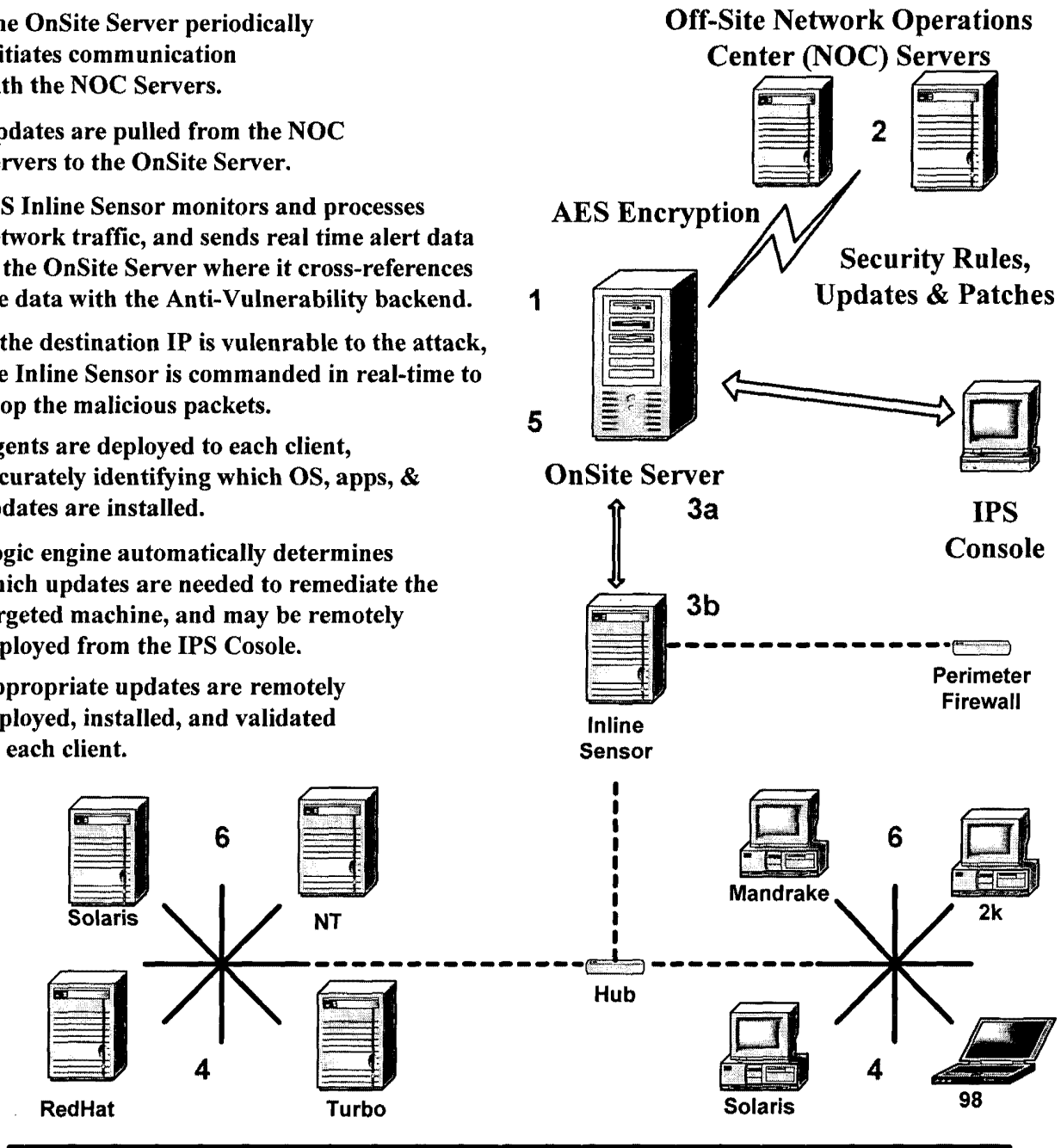
- 3. Remotely remediates the vulnerability.** When a machine is identified as vulnerable to an attack, an incident, a remediation function is also provided to the administrator to remotely deploy the appropriate update to the machine or device, install it, verifies the files and hashes, and documents it with change tracking and audit trail reports.

- **Supports Windows and UNIX variants**
- **Determines in real-time if targeted machine is vulnerable to an attack, virtually eliminating false positives**
- **Inline Sensor flex response terminates the attack**
- **Remotely patches targeted vulnerabilities in one click**
- **Installs in minutes**

Attacks are accurately identified and mitigated before they reach their targets, and targeted vulnerabilities may be remotely remediated. Therefore, complex threats are more effectively and efficiently mitigated. Correspondingly, the management and maintenance burden on administrators is reduced, saving time and effort.

Intelligent IPS Configured Network Overview Diagram

- 1 The OnSite Server periodically initiates communication with the NOC Servers.
- 2 Updates are pulled from the NOC Servers to the OnSite Server.
- 3a IPS Inline Sensor monitors and processes network traffic, and sends real time alert data to the OnSite Server where it cross-references the data with the Anti-Vulnerability backend.
- 3b If the destination IP is vulnerable to the attack, the Inline Sensor is commanded in real-time to drop the malicious packets.
- 4 Agents are deployed to each client, accurately identifying which OS, apps, & updates are installed.
- 5 Logic engine automatically determines which updates are needed to remediate the targeted machine, and may be remotely deployed from the IPS Cosole.
- 6 Appropriate updates are remotely deployed, installed, and validated on each client.



839 Main St. Suit 405 Lafayette, IN 47901-1403
 T I: 888-645-3676 Fax: 765-420-9256
 E-Mail: info@securityprofiling.com
 Web: www.securityprofiling.com

A white paper and trial download are available online at www.securityprofiling.com

PATENT APPLICATION SERIAL NO. _____

U.S. DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE
FEE RECORD SHEET

07/07/2003 AMABI1 00000054 60484085
01 FC:1005 160.00 DP

PTO-1556
(5/87)

07/01/03

17581 U.S. P.T.O.

227760 Please type a plus sign (+) inside this box

+

WEMMH SB/16 (4/03)

PROVISIONAL APPLICATION FOR PATENT COVER SHEET

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53(c).

INVENTOR(S)

Given Name (first and middle [if any])

Brett M.

Family Name or Surname

OLIPHANT

Residence

(City and either State or Foreign Country)

Lafayette, Indiana

☐ Additional inventors are being named on the _____ separately numbered sheets attached hereto**TITLE OF THE INVENTION (280 characters max)****SECURITY MANAGEMENT SYSTEMS AND METHODS**

Direct all correspondence to:

CORRESPONDENCE ADDRESS☐ Customer NumberPlace Customer Number Bar
Code Label Here

OR

Type Customer Number here

☒ Firm or
Individual Name

Matthew R. Schantz

Address **Woodard, Emhardt, Moriarty, McNett & Henry LLP**Address **Bank One Center/Tower, Suite 3700, 111 Monument Circle**City **Indianapolis** State **Indiana** ZIP **46204-5137**Country **U.S.A.** Telephone **(317) 634-3456** Fax **(317) 637-7561****ENCLOSED APPLICATION PARTS (check all that apply)**☒ Specification Number of Pages

35

☐ CD(s), Number☐ Drawing(s) Number of Sheets☐ Other☐ Application Data Sheet. See 37 CFR 1.76**METHOD OF PAYMENT OF FILING FEES FOR THIS PROVISIONAL APPLICATION FOR PATENT (check all that apply)**☐ Applicant claims small entity status. See 37 CFR 1.27.☒ A check or money order is enclosed to cover the filing fees☒ The Commissioner is hereby authorized to charge filing fees or credit any overpayment to Deposit Account Number:

23-3030

Filing Fee
Amount(\$)

\$160.00

☐ Payment by credit card. Form PTO-2038 is attached.

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.

☒ No.☐ Yes, the name of the U.S. Government agency and the Government contract number are: _____

Respectfully Submitted,

Date

JULY 1, 2003

SIGNATURE

TYPED or PRINTED NAME **Matthew R. Schantz**TELEPHONE **(317) 634-3456**

REGISTRATION NO.

40,800

(if appropriate)

Docket Number:

36029-2

USE ONLY FOR FILING A PROVISIONAL APPLICATION FOR PATENT

This collection of information is required by 37 CFR 1.51. The information used by the public to file (and by the PTO to process) a provisional application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 8 hours to complete, including gathering, preparing, and submitting the complete provisional application to the PTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, Washington, D.C. 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Mail Stop Provisional Application, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

"Express Mail" label number EV232973025US

Date of Deposit July 1, 2003

I hereby certify that this correspondence is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR §1.10 on the date indicated above and is addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Name of person signing paper or fee