Comparison of File Integrity Monitoring (FIM) Techniques for Small Business Networks

Brittany Wilbert
Department of Computer Science
Sam Houston State University
Huntsville, Texas, United States
BMW005@SHSU.EDU

Lei Chen
Department of Computer Science
Sam Houston State University
Huntsville, Texas, United States
LXC008@SHSU.EDU

Abstract—**File Integrity Monitoring (FIM) can provide the ability to track changes which have been made to an operating system or software as a result of malicious behavior. For small business environments, the need for these tools is present; however, because of the difficulty of managing such software many businesses may ignore using them. This paper will discuss how a small business should create criteria for comparing file integrity monitoring tools in order to locate the most effective product for their environment.**

Keywords-log file integrity monitoring; small business environments; auditing; log management

I. Introduction

Small business networks are susceptible to changes to the integrity of their files and documents due to potential compromises caused by malicious or naïve users, or malicious software (such as viruses and worms). As a result of these potential situations, individuals tasked with monitoring small business networks need the ability to monitor when there have been changes to major files and folders within individual workstations. FIM allows for changes to files and data to be tracked in case of modifications.

File integrity monitoring is a critical requirement for small businesses. Integrity monitoring allows for small businesses to ensure that their data is not being modified without their knowledge by either malicious outside actors or insiders. File integrity monitoring also supports changes to the environment to be thoroughly tracked and potentially reversed if combined with solutions such as system backups.

Our research creates a set of criteria which can be used to compare file integrity monitoring tools on the market. First, the necessity of file integrity monitoring will be discussed. Next, techniques used for integrity monitoring and theorized by previous works will be deliberated. After that, a framework for comparing file integrity monitoring software will be discussed, which can be used by small businesses to compare products for their environments. Finally, recommendations will be provided in regards to potential research topics relevant to file integrity monitoring.

II. Background

Small businesses are more susceptible to attacks by either external threats or internal users than ever before. In Verizon's 2013 Data Breach Investigation Report, 72% of small organizations (defined as having fewer than 1,000 employees) that reported to Verizon in 2012 were affected with hacking incidents and 54% were affected by malware [1]. These small to medium sized businesses are also susceptible to internal threats to their environment. These companies are particularly prone for attack because many of them do not own the monetary resources available to larger companies to expand their security to acceptable levels [2]. As a result, small businesses must be able to identify security solutions for their environment while being able to determine the functionality which are necessary to provide the best protection to their environment at a low cost.

For small businesses, a loss of data due to malicious incidents can lead to business failure or significant downtime. As a result of this critical risk, small businesses must be aware when changes to their data infrastructure have been affected as soon as possible. With the application of file integrity monitoring within a small business, it will help ensure that malicious and accidental/non-preventative changes (such as data corruption or user error) can be captured as early as possible.

Small businesses are also liable to both compliance and industry regulations as well as following best industry practices. For instance, for small businesses that make financial transaction, PCI-DSS Version 3.0 (Requirement 10.5.5, 10.6.1, and 11.5) [3] states that businesses need implement file integrity monitoring software to protect the Card Holder Environment (CHE) that contains credit card information. For other businesses, file integrity monitoring can be used to assist in meeting best practices and guidelines such as the National Institute of Standards and Technology (in Special Publication 800-53) [4] and the International Organization of Standards (ISO)/International Electrotechnical Commission (IEC) 27001 [5][6].

File integrity monitoring is one of the solutions which must be implemented in order to provide security to an environment. Monitoring changes to the state of the different software that has been deployed within an environment by using a data 'fingerprint' file is absolutely necessary to provide visibility into potential compromises that may have occurred within the environment. However, as a result of the difficulty that can emerge as a result of putting into place a file integrity solution, many small businesses forgo this process, and may as a result lose one way to determine when a compromise has occurred and how the system was compromised.

There are several reasons why file integrity monitoring solutions may not be deployed within a small business environment.

- Cost: Although there is now an increasing trend of small businesses providing funding to their information security departments [7], security solutions, including file integrity monitoring tools, may not be completely deployed because of the monetary cost associated with deploying security solutions. Since many small businesses attempt to run as cost effectively as possible until they are fully funded or bought out by a larger corporation, the desire of deploying an FIM solution may become a low priority. Also, many of these businesses, such as 'mom and pop' shops or other small establishments, most likely would not have the man-power necessary to deploy an FIM solution to their environments.
- "Good Snapshot": Most current generation file integrity monitors run on the assumption that the first snapshot obtained by the software is the known "good" baseline of the system which the FIM is being installed on. However, the only best time which a system can be certain that it has not been compromised previously is during a fresh installation of an operating system and known uncompromised software. Many small businesses may not have installed an FIM before allowing employees to access their network. However, with 87 percent of small businesses not having internet policies or best practices in late 2012 [8], the chance of an employee accidently (or purposely) exposing malware into their environment increases their security risks.
- "Security by Obscurity": Many individuals may believe that their business will not be a target of attacks because there are larger companies that can be compromised. However, as discussed earlier, this is not the case.

III. Literature Review of File Integrity Monitoring Techniques

Determining what level or levels that file integrity monitoring tools should monitor data is one of the first aspects which must be considered. Depending on the type of solution deployed, the solution may or may not monitor activity on all points which may be necessary to determine if unauthorized changes have been made. As a result, the need to discuss some of the different facets that FIM can assist in monitoring file integrity can assist in determining what tool or tools could best fit the environment, or system which the tool is being deployed.

Since changes to software on a system could occur relatively rapidly (such as software update and patches on individual workstations) or relatively slowly (such as the need for stable updates on servers), file integrity monitoring on this level needs to me maintained on multiple levels.

To organize the comparison, the sections have been arraigned to follow how implementation may occur from the kernel level, then discusses operating system, virtual machines, and cloud computing modules monitoring, and finally discusses monitoring of other types of software applications (such as email).

A. Kernel Level FIM

Automatic kernel integrity checker for multi-core environments can also be used for file integrity monitoring. Because other solutions require either significant cost or overhead to implement, or require virtualization where there may be instances of the hardware not being compatible, Shimada and Nakajima suggest that an integrity monitor can be run on a virtualization level to test the integrity of kernel data structures [9]. Traditional forms of integrity monitors can trigger false negatives because a malicious program can be hidden within the kernel level of the target operating system. The OS could also be checked using remote monitoring solutions, such as a signature-based integrity monitor, however an administrator, the person who defined what is being scanned within an environment, can potentially ignore workstations that have been compromised [9].

The method provided by the authors to resolve these problems is an integrity checker which uses invariants of the data structure of the kernel. This structure allowed for a global variable that is used in a kernel as well as invariant that defines the limit that a variable can take to automatically generate integrity monitoring without defining security policies [9]. This activity is analyzed by a 'Daikon' tool Java programs. Since not all variables can be reviewed, the number of invariants has to be reduced [9]. Also, the tool is limited by how far it can traverse in the kernel, the directory of the Linux source code thread it can target, differences in invariants, and that not all variables are used in the kernel [9].

The discussion of kernel integrity monitoring is important in regards to file integrity monitoring, because if the kernel itself is compromised the entire system is essentially compromised as well. The use of invariants can be implemented as a potential means of creating a security policy dependent infrastructure for integrity monitoring that allows for its potential use in other monitoring within the computer's infrastructure.

B. Operating System Level FIM

Another file integrity monitoring technique, which can be used dependent of virtual machines, is the use of the operating system to provide protection. The use of a tool call XenFIT, allows for protection of computer systems using the operating system's own protections, in this case using the same operating system, Xen OS. This solution states that since the software is supposed to work natively with Xen environments, it is easy to maintain, tamper-resistant to attacks, and deployment is stealthy and can minimize suspicion from attacks [10].

Quynh and Takefuji, the authors of this design, discuss that well-known file integrity monitoring tools, such as Tripwire, use a baseline of activity that has to be assumed as safe, and if any modifications of data are detected, they are then alerted to the administrator [10]. The authors suggest that the problems with these solutions are that detection is delayed, and the complexity of environments that deploy the solutions has to be planned far in advance or they can fail [10]. Also, other negative factors include that maintenance of the solution is very high, information for many File Integrity Tool (FIT) solutions are unavailable or do not provide enough information for investigations, and finally if the solution is found within the environment the attacker can compromise it and hide their tracks to prevent its usefulness [10].

The solution provided for Xen is the use of XenFIT, which helps to assist with mitigating or removing many of the barriers that would prevent a FIT from being deployed. Their solution runs in real-time and natively to the Xen environment. This allows for activity to be monitored without exposing itself to the attacker. XenFIT uses breakpoints to determine when activity has occurred. It then uses correlation to review system-calls within the environment. This solution also uses a separate security policy per device to determine the attributes of the system locally instead of it being stored remotely. If a violation is found, XenFit can then use the DomU kernel member to obtain information about the activity [10].

Because of the structure of Xen OS environment, this type of FIT appears to have more readability implemented within the environment compared to other operating system. This approach to use a real-time FIT to review data can be seen similar to the VMGuard implementation discussed in a later section of this paper. However, the performance and security criteria that they used to test the software itself can be implemented to test other instances of FIT for their readiness within an infrastructure.

C. *Virtual Machines FIM*

1) Multi-byte No-Operation (NOP) Injection for File System Integrity

This technique, introduced by Kim et al., is discussed in relationship to the use of virtual machines. Virtual machines are used in many areas, including in small business environments, to reduce software deployment cost within an organization. Ensuring that businesses that deploy virtual machine environments do not become compromised is important.

This FIM design uses the implementation of the NOPFIT (NOP File System Integrity Tool). This is a debugger which stores a breakpoint into a temporary area [11]. The code is then monitored using the gcc compiler for NOP instruction. The lguest process thread is used to obtain information about the breakpoints, apply security policies, and gather NOPFIT processes [11]. Also, a Kernel object parser is used to determine when there has been any stack-pointer changes within the current kernel, and is also used to monitor activity. Finally two different FIT implementations (NOPFIT and INT3FIT) were compared and results indicated that the NOPFIT took approximately 900 ms less than the INT3FIT utilization [11].

The reason that Kim et. al. suggest a multi-byte NOP injection with a FIT is to provide an increased performance threshold for file integrity monitoring [11]. The authors argue that traditional FIT infrastructures installed must first have security polices predefined with a lower performance threshold due to the amount of time it takes to intercept system-calls. Because of this, by using this software breakpoint debugging technique, it can inject bytes of code into a running system at run time, exception handling, and use predefined virtual machine software, such as ‘lguest’ virtual machine and memory virtualization to conduct integrity monitoring.

2) Xen OS VMGuard File Integrity Monitoring

Another file integrity monitoring solution for virtual machines is the use of the operating system run by the virtual machine itself. Similar to XenFit discussed in the Operating System section, VMGuard is an integrity-monitoring tool that can be used to detect virtual machines for malicious activity such as rootkits and other threats. The creators of VMGuard suggest that privileged access to virtual machine’s management console must be monitored for activity, and that an attacker will typically attack the management console (Domain0 in Xen) using exploits designed to compromise privileges used on the console, and therefore obtain complete administrative access to the system [12]. Because of this, integrity monitoring tools for virtual machines must be designed to survive an attack even if the Domain0 area is compromised. VMGuard contributes integrity monitoring as well as an improved performance threshold that has explicitly been designed for Xen environments.

This design uses several key points to provide real-time monitoring while being tamper-resistant with integrity verification. To complete this process, the VMGuard architecture is designed to take a predefined policy and distribute it using GuardDomainU in trusted mode [12]. Once this is completed the administrator can begin a logger that transports the integrity measurements and a verifier to compare the logging records with the latest measurements taken from the environment. Their research included performance evaluation that was done using their VMGuard’s DomainUs and measured the execution time on three different machines as well as compared to an Network File System (NFS). It is concluded that VMGuard does have more initial overhead compared to DomainUs, however, performance loads improve compared to a NFS [12]. As a result, this allows for better virtual machine support while reducing the load that is created as a result of running the software.

D. *Cloud Computing FIM*

The next area of focus is the need for file access monitoring for cloud environments. Ko, Jagadpramana, and Lee discuss that some of the largest obstacles for cloud adaption is the need for trust, transparency and accountability of data stored within the environment [13]. Their solution to this problem, Flogger, is designed to detect activity within the kernel level, which is important for securing operations within the cloud.

The authors discuss that the provenance of data life cycles, auditable viewing of files, as well as efficient storage and analytics are necessary to determine when activity has occurred within a cloud infrastructure [13]. Flogger uses a Linux Loadable Kernel Module (LKM) and a Windows Device Driver to interpret file and network operations on the virtual machine level (LKM) as well as intercept file operations on the physical machine level (Windows Device Driver). Flogger captures all file-centric log messages and compared the results of previous activity within the virtual machine and physical machine levels.

Flogger is tested based on several use case scenarios to determine its effectiveness. Examples include determining unauthorized user file access, capturing file transfers on a network, as well as obtaining provenance data from the log messages received [13]. One of the examples is the use of Flogger to compare the security level of the physical machine and the underlining secure copy protocol (SCP) activity of the machine with the log messages that are being created on the virtual machine level. Because this activity is correlated, it can be used to provide additional information that can be used for forensics and analytics [13].

Because the solution implements activity that is seen on both parts of a cloud environment, it is able to provide multiple examples of activity for investigation. Although some of the use cases are specific to virtual machines, altered forms of these tests can be used for other implementations of integrity monitoring.

E. Miscellanous Software FIM

File integrity monitoring can also be implementing on other types of software infrastructures. Providing protection on this level allows for data integrity to be applied across the OS Model.

1) FIM for Storage Data Integrity

Storage integrity monitoring is another aspect of file integrity which has not received much discussion. Sivathanu, Wright, and Zadok provide a novel technique which uses “logical redundancy” to provide integrity monitoring [14]. The authors posit that integrity violations can be due to both storage errors and malicious intrusions that occur within an environment [14]. As a result of these actions, the need for ‘integrity assurance’ for storage device is necessary to provide an integrity check of the performance, security and other factors. Also these checks can be used to detect integrity violations.

There are several use cases as well as examples of activity which can occur resulting in a compromised storage environment. These variables include malicious intrusions, hardware and software errors, as well as accidental user errors [14]. The authors have concluded that most of these methods are not sufficient for storage integrity monitoring because they lack in some key area that needs to be monitored compared to other services outside of the storage component itself. To argue this statement, the authors suggest that the technique of logical redundancy can be used to mitigate current integrity monitoring failures. This technique exploits the semantic redundancy that occurs inherently within files. This method uses a file system called “Pilot” which maps and updates activity that occurs on the storage device [14]. It then uses file system bitmaps and on-disk data structures to perform integrity checking to detect corruption to the pointers that are within the storage device [14].

This technique is another method of determining file system integrity, by using the storage which is holding the files to determine when changes have occurred to files within the system. Because it involves an entire operating system to be installed on top of whatever storage that can be used, it may cause an additional burden on administrators to implement this type of technique on a preexisting infrastructure. However, if the environment is a new environment, this technique maybe an interesting solution to file integrity monitoring needs.

2) Email Level FIM through Human Behavior Monitoring

The next technique is the use of human behavior to determine when FIM solutions should be reviewed. Sasaki discusses an initial system architecture that can alert when there is a data leak within an environment. This technique uses a mixture of access control as well as anomaly detection to determine when data leakage has occurred as well as what access changes or privileges are given to insiders to determine when these changes occur [15]. Sasaki then gives two problems that are currently present in regards to determining if there has been any malicious activity from insiders: 1) users may be given additional privileges which can enabled them to conduct malicious activity, and 2) detection of this activity is difficult because it cannot just be determined by changes to everyday operations [15].

The framework suggested to bridge these problems is to trigger based on activity which is done by malicious insiders, monitor the activity made by insiders, and then provide analytical tools to review the activity made by the trigger [15]. As a result, if a malicious insider attempts to hide their tracks by deleting activity, the system will be able to monitor additional activity that may be performing in order to provide more evidence. To do this, a file and email monitor is used to record file deletions within a system. The file monitor then records times that the activity occurred as well as backs up the data to a remote system [15]. This architecture also uses a log store as well as an analyzer to compare the files and emails that have been modified during a period of time and calculate a score based on these changes to the system and provide a trigger for investigation.

The author’s main goal is to provide a structure in which tools can be implemented for monitoring suspicious activity for insiders within a company or organization. Besides just monitoring for outsider threats using a file integrity monitor, these same practices can be used to detect insider activity which is occurring within the environment that can cause compromise.

IV. Combining FIM With Logging Solutions

To further leverage file integrity monitoring, the use of log management solutions is vital to ensure that the data being collected by the FIM is monitored. A log management solution should allow for different types of log messages to be implemented as well as to provide better accountability into the activity seen within the environment [16]. When considering combining an FIM with a log management solution, the following considerations should be made:

- Performance testing of the log monitoring solution must be tested before implementation [17] to provide feedback on how the log management acts under stress that the FIM will bring.

- The network environment, topology, and structure in which the log management solution will be deployed within should be considered [17]. This is to ensure that enough hardware space is allocated for logs that are retrieved from computer sources.
- A log management solution should use a standardized programming solution to provide better log management and collection [17]. This will also allow determining if the FIM deployed will be compatible with the log management solution.
- If the log management solution does include FIM capability, further investigation should be made to determine if the software provides the best coverage for the small business's needs. An FIM should not be automatically turned on unless this testing is completed.

V. PROTECTING FIM DATA

The protection of FIM fingerprint data must also be discussed because if the data is not protected, the FIM will become unreliable. An example protection mechanism that can be placed within a distributed file system is a cryptographic access control for the system [18]. This mechanism can be used within an environment that is not trusted or if it lacks defined controls. Because of the use of the Internet and other untrusted areas, administrators need to monitor activity within trusted spaces in their environment as well as untrusted spaces [18]. Typically reference monitoring is used, which defines by default that the areas that the rights are being references are safe before it is being monitored [18]. However there are instances where the space being referenced should not use reference monitoring because there is a lack of controls that can be implemented as a result.

Because of these unsafe areas, using a log-structured file system where modifications are written to a log file instead of directly onto the system itself, and using a cryptographic access control mechanism can be used to define integrity and availability of access control lists. However, the use of log files can provide a burden to the system due to the increasing size of log files. Because of this, optimization of the files through compression and intelligent read and write ability need to be used to reduce the size of the log file [18].

Another difficulty is determining if the files uphold confidentiality, integrity and availability. The infrastructure of the implementation uses cryptography in a Cryptographic NFS (CNFS). To show this, Twofish cipher is used as a symmetric cipher and a MD5 hash is also used to verify that the data is valid [18]. This technique can then be used to determine if the FIM fingerprint files have been compromised themselves.

VI. CONSIDERATIONS FOR SELECTING AN FIM

Each of the file integrity monitoring techniques and solutions discussed exhibit different attributes which can be leveraged towards the improvement of protecting data within the small business environment. However, each solution does have limitations and should be individually reviewed depending on the small business environment.

For instance, a small business that uses physical servers and not virtual machines would not require many of the methods used for protecting virtual machines. However, a small business should make sure the following requirements are met when researching any file integrity monitoring tools:

- FIM solutions should provide visibility as well as alerting of changes to the system it has been placed in.
- A reliable FIM should prevent tampering of its own file, or alert on when the software's files have been tampered.
- FIM 'fingerprints' that are created by a FIT should be resistant to tampering. The FIM should provide guidance to how these records should be handled.
- The fingerprint files should have strong encryption and should be stored at rest within a safe location.
- Combining an FIM with a log management system will allow for a comprehensive look into the activity that has occurred within the environment, and when the activity occurred, as well as what occurred.
- A potential centralized file integrity management tool should ensure that it can be protected from tampering.
- Any solution should be reviewed by security and operational teams that will be managing the solution as well as subject matter experts within the organization.

A. *FIM Techniques*

Techniques which should be considered when looking for a solution include:

- Masking of activity: Minimize, wherever possible, what processes, services and other activity to prevent malicious users from initially detecting the FIM use within the environment.
- Self-protecting: Besides the organization hardening the software, the FIM should have some capability to protect and alert when the product has been tampered with or before attempts of disabling occur.
- A review of fingerprints should be performed before and after any 1) patches, 2) system configuration changes 3) any time there has been an authorized change within the environment.
- Detection of patches: Anytime a change occurs within a FIM protected environment, including standard patches, it should alert. This should also indicate that a new baseline which includes the patched activity should be conducted.
- Alerting capabilities: Setting up and configuring alerting is an important aspect of implementing a successful FIM. A business must ensure that alerting is enable for the infrastructure that the FIM is installed to, as well as alert/log anytime there are changes to the software itself.
- Reporting: Management and operational reports of activity may be available on the FIM protect. Although not as important as other techniques, reporting allows for long-

term trends to be possibly detected the longer the FIM and other security tools are deployed within the environment.

B. *Possible FIM Solutions*

- A few FIM solutions a business may consider include, but are not limited to, the following (in no particular order):
- Tripwire: Supports most operating systems, such as Windows, Linux, and Unix systems, as well as other platforms (for example databases and virtual environments) [19]. Creates configuration assessment policies created using the baseline fingerprint created at first run, then alerts when there are modifications to the baseline. This product has been used by a number of businesses for several years.
- LogRhythm File Integrity Monitoring: Supports Windows, Unix and Linux systems. Is also integrated with LogRhythm's log management and System Information and Event Manager (SIEM) [20]. If a business is looking for one product that supports multiple information security requirements, this product may
- Trustwave File Integrity Monitoring states that its "Windows-based" software supports "POS [Point of Sales] devices, laptops, desktops and servers" [21]. This could be an option for small businesses that conduct credit card transactions.
- AlienVault File Integrity Monitoring tool is integrated into their United Security Management (USM) tool, which includes an intrusion detection system (IDS) [22]. This tool provides coverage for environments that are based in the cloud and other virtual environments. AlienVault's documentation states that it supports PCI-DSS 2.0 requirement 11.5 [22]. However, this is a younger company and this could be either a positive or negative business consideration.
- McAfee Integrity Monitoring: A potentially more expensive solution that can be used for databases and network devices [23]. However, McAfee provides licenses based on the number of nodes that will deploy the system, which can allow for businesses to give forecasts based on future business requirements. This product also includes the ability to more easily integrate the solution with the McAfee anti-virus solution and management tools/console for businesses that either already have the product deployed or may require a new anti-virus vendor.

All five (5) solutions potentially provide multiple software solutions outside of the file integrity monitor product. This may be an incentive for small businesses that need to quickly create a layered security deployment solution. However, all small businesses should be prepared to thoroughly review and test all potential solutions that are deployed, and not just isolate testing to just a FIM product. This testing should be performed for new and existing software within the business environment to make sure the new security solution(s) deployed are compatible and are best for business. If this is not performed, unintended consequences such as server or workstation downtime due to misconfiguration of software solutions may result. These consequences can lead to a business spending beyond their budget in both personnel and monetary resources in order to remediation the problems.

VII. FIM Deployment Strategy for Small Businesses

Deploying a file integrity monitoring software involves many steps to make sure that the process is done correctly. This involves the inclusion of employees, management, and administrators who are charged with adding the solution to the field. The following critical components should be included when considering an FIM tool:

A. *Review of Who and What is On the Network*

- Before the file integrity monitoring software is implemented, the system must be ensured to be in its best "good" state. The system should be scanned for malware and other malicious software. If a compromised is found in this step, mitigation should be performed for the system immediately.
- Everyone at the company should have up-to-date security training. This process will reduce inappropriate use of the work network or at least inform employees of the possible consequences of their actions. This process should be frequently repeated because of the unpredictability of employees and changes to social engineering techniques.
- Security, operations, and IT teams involved with deploying the solution must, at a minimum, be involved in the discussion and review of potential FIM solutions investigated before deployment. They must also be further trained in how the software works before, during and after deployment of the solution.

B. *Enabling FIM Within the Environment*

- File integrity monitoring should be enabled on all vital, mission critical systems on a network.
- The FIM's files should be updated every time there is a change to the software or hardware on the system. Without an update, the fingerprint for the system will not be correct if a compromised occurs.
- The FIM fingerprint should be backed up and stored every time it is updated to an external device.
- The FIM fingerprint files should be backed up to multiple outside sources.
- If monetary resources are available, one of the FIM fingerprint files should be backed up and stored in a location outside of the home or office in case of disaster recovery.

VIII. Conclusions

There are more opportunities available to research how small businesses can protect themselves from malicious activity. Since malicious individuals target these environments to provide further space to potentially compromise larger, more lucrative targets and assets, the need for better education and prevention of attacks on small businesses is necessary. This is also necessary to be compliance with industry requirements as well as to practice best security standards. Although file integrity monitoring does allow better visibility into these potential compromises, because of the barrier to entry for these products many small businesses do not take the steps necessary to implement this type of solution.

Since the largest barrier of entry is the cost associated with deploying such solutions, research into developing easier, more user-friendly FIM solutions would allow for more small businesses to deploy these products. Also, allowing for these products to be less cost preventative will allow businesses who are selling these products a wider customer base than just selling to customers who have the finances to purchase more expensive solutions. Finally, implementing an FIM into a log management solution will allow small businesses to leverage multiple points of information when obtain data about the health of their environment.

A combination of the techniques discussed would be the best solutions to providing an FIM solution to small businesses. Ultimately, combining multiple solutions can increase the cost of the product itself. Overall, to provide the best FIM solution, a small business must be able to locate a product which provides the most options without overburdening their infrastructure with intensive, hardware burdening software. By determining the type of environment, as well as how the software will be deployed, small businesses will be able to determine the best solution that is needed for the type of environment they are currently running.

REFERENCES

[1] “2013 Data Breach Investigations Report,” © 2013 Verizon Enterprises. Available: http://www.verizonenterprise.com/resources/reports/rp_data-breach-investigations-report-2013_en_xg.pdf.

[2] Gonsalves, A., “Hackers increasingly zero in on small businesses, Symantec says,” CSO: Security and Risk, Aug 2012. [Online]. http://www.csoonline.com/article/712942/hackers-increasingly-zero-in-on-small-businesses-symantec-says.

[3] PCI Security Standards Council, “PCI-DSS 3.0,” © 2014 PCI Security Standards Council, LLC.. [Online]. Available: https://www.pcisecuritystandards.org/security_standards/documents.php.

[4] NIST, “Security and Privacy Controls for Federal Information Systems and Organizations: Special Publication 800-53,”4th rev. National Institute of Standards and Technology, 2013. doi:10.6028/NIST.SP.800-53r4.

[5] “ISO/IEC 27001:2013,” © 2013 ISO/IEC. [Online] Available: http://www.iso.org/iso/catalogue_detail?csnumber=54534.

[6] Disterer, G., “ISO/IEC 27000, 27001 and 27002 for Information Security Management,” In *Journal of Information Security*, Apr 2013, pp. 92-100. 42011 [Online]. Available: http://dx.doi.org/10.4236/jis.2013.

[7] “Insights on governance, risk and compliance: EY’s Global Information Security Survey 2013,” Ernst & Young, Oct 2013. [Online]. Available: http://www.ey.com/Publication/vwLUAssets/EY_-_2013_Global_Information_Security_Survey/$FILE/EY-GISS-Under-cyber-attack.pdf.

[8] “2012 National Small Business Study,” National CyberSecurity Alliance, Symantec, and JZ Analytics, Oct 2012. [Online]. Available: http://www.staysafeonline.org/download/datasets/4393/2012_ncsa_symantec_small_business_study_fact_sheet.pdf.

[9] Shimada, H. and T. Nakajima, “Automatic Invariant Generation for Monitoring OS Kernel Integrity,” In 2012 IEEE 18th Int. Conf. on Embedded and Real-Time Computing Systems and Applicat. (RTCSA 2012), IEEE August 2012, pp. 408-410.

[10] Quynh, N. A. and Y. Takefuji. “A novel approach for a file-system integrity monitor tool of Xen virtual machine,” In Proc. of the 2nd ACM Symp. on Inform., Comp. and Commun. Security (ASIACCS '07), Robert Deng and Pierangela Samarati (Eds.). ACM, 2007, pp. 194-202. Available: DOI=10.1145/1229285.1229313 http://doi.acm.org/10.1145/1229285.1229313.

[11] Kim, J., I. Kim and Y.I. Eom, “NOPFIT: File System Integrity Tool for Virtual Machine Using Multi-byte NOP Injection,” In Computational Sci. and its Applicat., Int. Conf. IEEE, March 2010, pp. 335-338.

[12] Fang, H., Y. Zhao, H. Zang, H. H. Huang, Y. Song, Y. Sun and Z. Liu, “VMGuard: An Integrity Monitoring System for Management Virtual Machines,” In Int. Conf. on Parallel and Distributed Syst. IEEE, December 2010, pp. 67-74.

[13] Ko, R. K. L., P. Jagadpramana and B.S. Lee, “Flogger: A File-Centric Logger for Monitoring File Access and Transfers within Cloud Computing Environments.” In Int. Joint Conf. of IEEE TrustCom/IEEE ICESS/FCST. IEEE, November 2011, pp. 765-771.

[14] Sivathanu, G., C. P. Wright and E. Zadok, “Ensuring data integrity in storage: techniques and applications,” In Proceedings of the 2005 ACM Workshop on Storage Security and Survivability (StorageSS '05). © ACM, 2005, pp. 26-36. [Online]. Available: http://doi.acm.org/10.1145/1103780.1103784.

[15] Sasaki, T., “Towards Detecting Suspicious Insiders by Triggering Digital Data Sealing,” In Int. Conf. on Intelligent Networking and Collaborative Syst., IEEE, December 2011, pp. 637-642.

[16] Casey, D, “Turning log files into a security asset,” Network Security, Volume 2008, Issue 2, ppg. 4-7, © 2008 Elsevier Ltd. doi: 10.1016/S1353-4858(08)70016-3.

[17] Mounji, A., B. Le Charlier and D. Zampuniéris, “Distributed audit trail analysis,” Symp. on Network and Distributed System Security, pp. 102-112, © Feb 1995 IEEE. doi: 0-8186-7027-4/95.

[18] Harrington, A. and C. Jensen, “Cryptographic access control in a distributed file system,” In Proc. of the Eighth ACM Symp. on Access Control Models and Technologies (SACMAT '03). ACM, 2003, pp. 158-165. Available: DOI=10.1145/775412.775432, http://doi.acm.org/10.1145/775412.775432.

[19] Tripwire, “Supported Platforms & Devices,” © 2014 Tripwire, Inc.. [Online]. Available: http://www.tripwire.com/it-security-software/scm/specifications/supported-devices/tripwire-enterprise-hardware-configuration-parameters/showMeta/2/.

[20] LogRhythm, “File Integrity Monitoring.” © 2013 LogRhythm, Inc.. [Online]. Available: http://www.logrhythm.com/siem-2.0/features-components/file-integrity-monitoring.aspx.

[21] Trustwave, “Trustwave File Integrity Monitoring (FIM),” © Trustwave Holdings, Inc.. [Online]. Available: https://www3.trustwave.com/file-integrity-monitoring.php.

[22] AlienVault, “File Integrity Monitoring: What It Does and Why You Need It,” © 2014 AlienVault, Inc.. [Online]. Available: http://www.alienvault.com/solutions/pci-dss-file-integrity-monitoring.

[23] McAfee, “McAfee Integrity Monitor,” © 2014 Intel Corporation. [Online]. Available: http://www.mcafee.com/us/products/integrity-monitoring-for-databases.aspx.