PCI DSS File Integrity Monitoring Explained

NNT™
SECURITY THROUGH SYSTEM INTEGRITY
NOW PART OF netwrix

A New Net Technologies Whitepaper

Mark Kedgley

CTO - New Net Technologies

www.newnettechnologies.com

Abstract

Although FIM or File-Integrity Monitoring is only mentioned specifically in two sub-requirements of the PCI DSS (10.5.5 and 11.5), it is actually one of the more important measures in securing business systems from card data theft.

What is it, and why is it important?

File Integrity Monitoring systems are designed to protect card data from theft. The primary purpose of FIM is to detect changes to files and their associated attributes.

However, this article provides the background to three different dimensions to File Integrity Monitoring, namely:

- Secure hash-based FIM, used predominantly for system File Integrity Monitoring
- File contents integrity monitoring, useful for configuration files from firewalls, routers, and web servers
- File and/or folder access monitoring, vital for protecting sensitive data

> *“...anti-virus defenses are typically only aware of 62% of the world’s malware...”*
>
> *http://doi.ieeecomputersociety.org/10.1109/MC.2010.187*

How far should FIM measures be taken? Start with System Files...

As a starting point, it is essential to monitor the Windows/System32 or SysWOW64 folders, plus the main Card Data Processing Application Program Folders.

For these locations, running a daily inventory of all system files will detect all additions, deletions, and changes. Harnessed correctly, this data will provide an ‘at a glance’ report from which potential security breaches can be identified. Additions and Deletions are relatively straightforward to evaluate, but how should changes be treated, and how do you assess the significance of a subtle change, such as a file attribute change? The answer is that ***ANY*** file change in these critical locations must be treated with equal importance. Most high-profile PCI DSS security breaches have been instigated via an ‘inside man’ – typically a trusted employee with privileged admin rights. For today’s cybercrime there are no rules.

The industry-acknowledged approach to FIM is to track all file attributes and to record a secure hash. Any change to the hash when the file-integrity check is re-run is a red alert situation – using SHA1 or MD5, even a microscopic change to a system file will denote a clear change to the hash value. When using FIM to govern the security of key system files there should never be any unplanned or unexpected changes – if there are, it could be a Trojan or backdoor-enabled version of a system file.

This is why it also crucial to use FIM in conjunction with a ‘closed loop’ change management system – planned changes should be scheduled and the associated File Integrity changes logged and appended to the Planned Change record.

Secure Hash Based FIM

Within a PCI DSS context, the main files of concern include System files e.g. anything that resides in the Windows/System32 or SysWOW64 folder, program files, or for Linux/ Unix, key kernel files.

The objective for any hash-based File Integrity Monitoring system as a security measure is to ensure that only expected, desirable, and planned changes are made to in scope devices. The reason for doing this is to prevent card data theft via malware or program modifications.

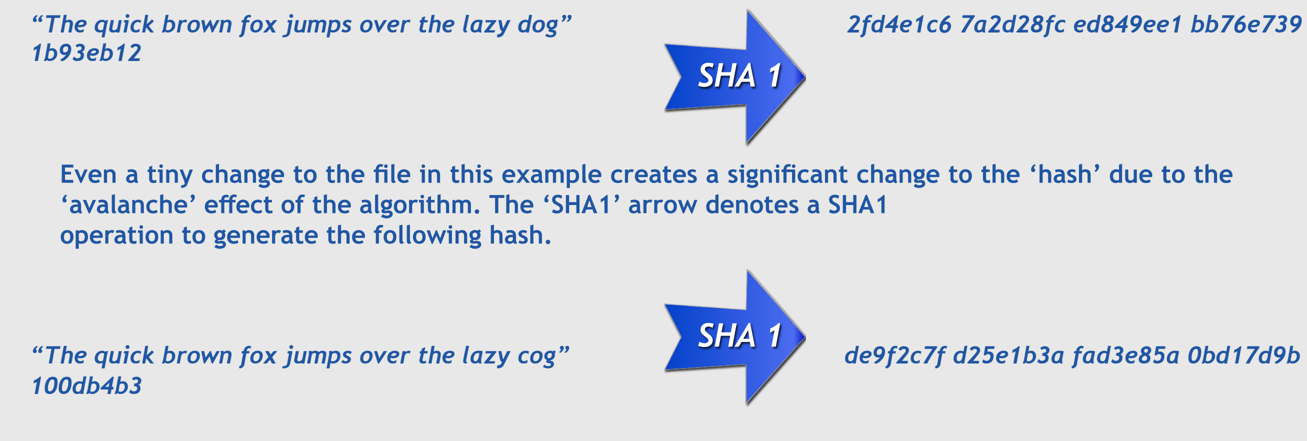
> “ ...For today’s cybercrime there are no rules.. ”

Imagine that a Trojan is installed onto a Card Transaction server – the Trojan could be used to transfer card details off the server. Similarly, a packet sniffer program could be located onto an EPoS device to capture card data – if it was disguised as a common Windows or Unix process with the same program and process names then it would be hard to detect.

For a more sophisticated hack, what about implanting a ‘backdoor’ into a key program file to allow access to card data?

These are all examples of security incidents where File-Integrity Monitoring is essential in identifying the threat. Remember that anti-virus defenses are typically only aware of 62% of the world’s malware (see http://doi.ieeecomputersociety.org/10.1109/MC.2010.187 for a recent study) and an organization hit by a zero-day attack (zero-day marks the point in time when a new form of malware is first identified – only then can a remediation or mitigation strategy be formulated but it can be days or weeks before all devices are updated to protect them).

The diagram in Figure 1 shows how the SHA1 secure hash algorithm generates a distinctly different hash value even for the smallest change to the data within a file. This provides a unique means of verifying that the integrity of a file has been maintained.



“The quick brown fox jumps over the lazy dog” *2fd4e1c6 7a2d28fc ed849ee1 bb76e739*
1b93eb12

Even a tiny change to the file in this example creates a significant change to the ‘hash’ due to the ‘avalanche’ effect of the algorithm. The ‘SHA1’ arrow denotes a SHA1 operation to generate the following hash.

“The quick brown fox jumps over the lazy cog” *de9f2c7f d25e1b3a fad3e85a 0bd17d9b*
100db4b3

Figure 1 - Illustration of how a secure hash algorithm creates a unique ‘hash’ based on the contents of a file

File Content and Configuration File Integrity Monitoring

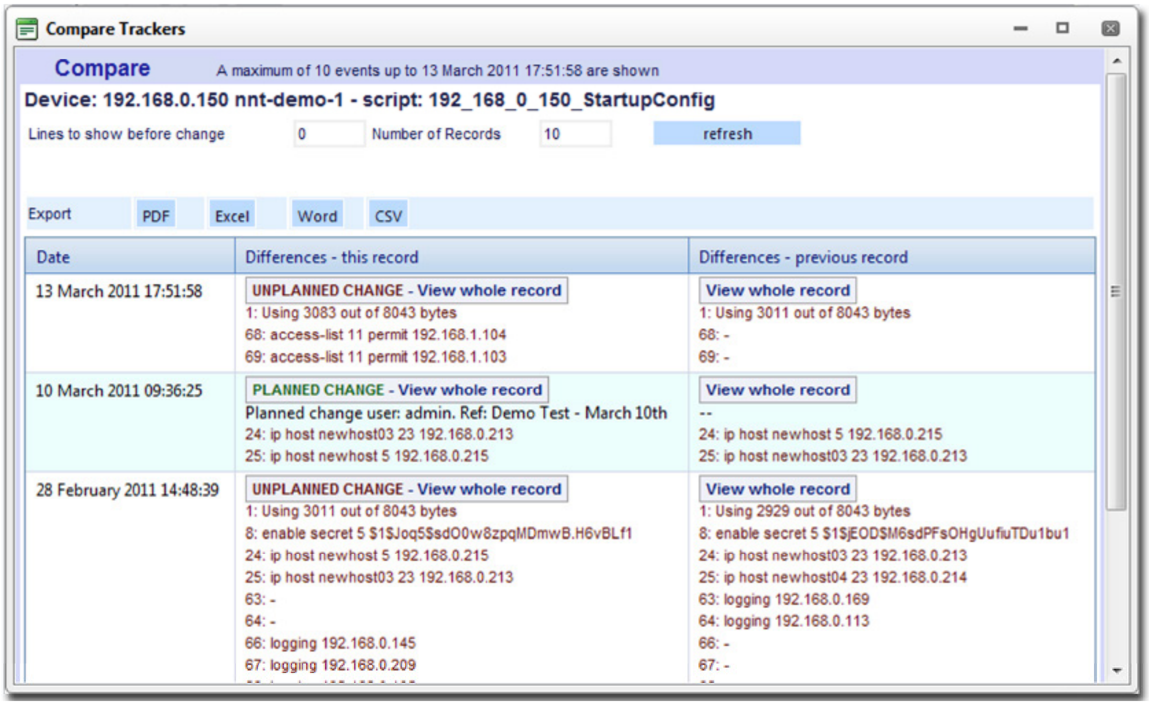
Whilst a secure hash checksum is an infallible means of identifying system file changes, this does only tell us that a change has been made to the file, not what the actual detail of the change is.

Sure, for a binary-format executable, this is the only meaningful way of conveying that a change has been made, but a more valuable means of File Integrity Monitoring for 'read-able' files is to keep a record of the file contents. This way, if a change is made to the file, the exact change made to the readable content can be reported.

> *“6.4 Follow change control procedures for all changes to system components”*
>
> *https://www.pcisecuritystandards.org/*

For instance, a web configuration file (php, aspnet, js or javascript, XML config) can be captured by the FIM system and recorded as readable text; thereafter, changes will be detected and reported directly.

Similarly, if a firewall access control list was edited to allow access to key servers, or a Cisco router startup config altered, then this could allow a hacker all the time needed to break into a card data server.



A maximum of 10 events up to 13 March 2011 17:51:58 are shown

Device: 192.168.0.150 nnt-demo-1 - script: 192_168_0_150_StartupConfig

Lines to show before change: 0
Number of Records: 10

Date	Differences - this record	Differences - previous record
13 March 2011 17:51:58	UNPLANNED CHANGE - View whole record 1: Using 3083 out of 8043 bytes 68: access-list 11 permit 192.168.1.104 69: access-list 11 permit 192.168.1.103	View whole record 1: Using 3011 out of 8043 bytes 68: - 69: -
10 March 2011 09:36:25	PLANNED CHANGE - View whole record Planned change user: admin. Ref: Demo Test - March 10th 24: ip host newhost03 23 192.168.0.213 25: ip host newhost 5 192.168.0.215	View whole record -- 24: ip host newhost 5 192.168.0.215 25: ip host newhost03 23 192.168.0.213
28 February 2011 14:48:39	UNPLANNED CHANGE - View whole record 1: Using 3011 out of 8043 bytes 8: enable secret 5 $1$Joq5$sdO0w8zpqMDmwB.H6vBLf1 24: ip host newhost 5 192.168.0.215 25: ip host newhost03 23 192.168.0.213 63: - 64: - 66: logging 192.168.0.145 67: logging 192.168.0.209	View whole record 1: Using 2929 out of 8043 bytes 8: enable secret 5 $1$jEOD$M6sdPFsOHgUufiuTDu1bu1 24: ip host newhost03 23 192.168.0.213 25: ip host newhost04 23 192.168.0.214 63: logging 192.168.0.169 64: logging 192.168.0.113 66: - 67: -

One final point on file contents integrity monitoring - Within the Security Policy/Compliance arena, Windows Registry keys and values are often included under the heading of FIM. These need to be monitored for changes as many hacks involve modifying registry settings. Similarly, a number of common vulnerabilities can be identified by analysis of registry settings.

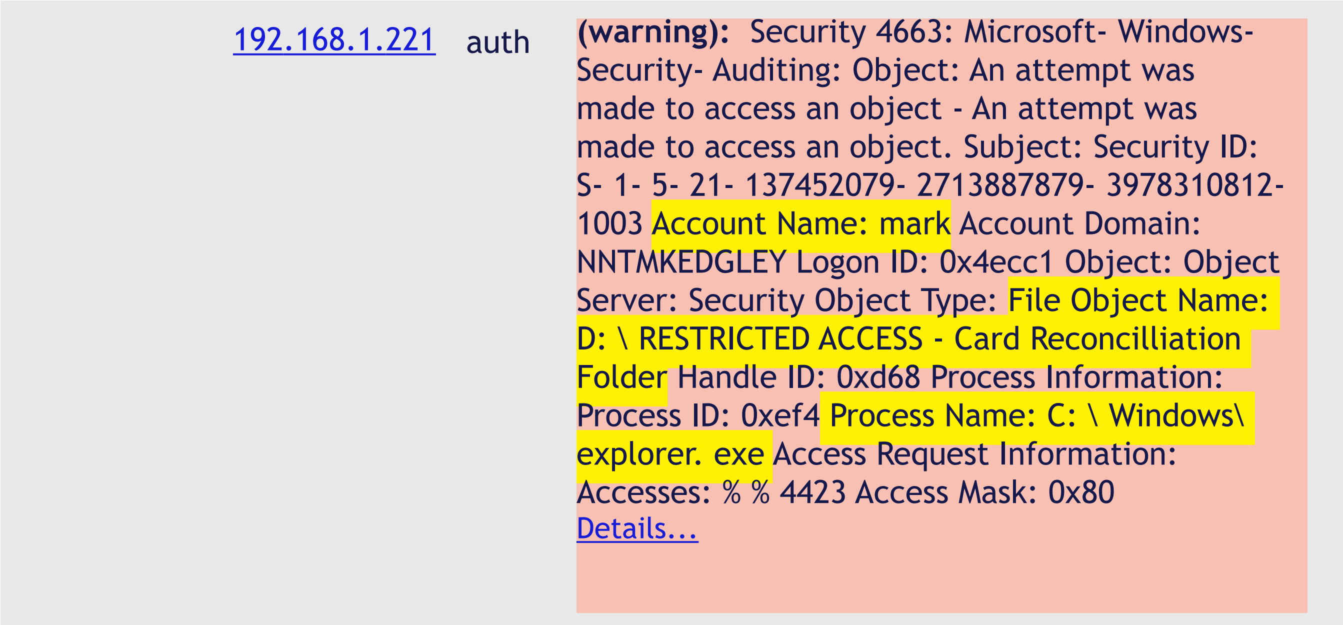
File and/or Folder Access Monitoring

The final consideration for File Integrity Monitoring is how to handle other file types not suitable for secure hash value or contents tracking. For example, because a log file, database file etc will always be changing, both the contents and the hash will also be constantly changing. Good file integrity monitoring technology will allow these files to be excluded from any FIM template.

However, card data can still be stolen without detection unless other measures are put in place. As an example scenario, in an EPoS retail system, a card transaction or reconciliation file is created and forwarded to a central payments server on a scheduled basis throughout the trading day. The file will always be changing – maybe a new file is created every time with a time stamped name so everything about the file is always changing.

Standard practice would be to place the file in a secure folder on the EPoS system to prevent user access to the contents. However, an 'inside man' with Admin Rights to the folder could view the transaction file and copy the data without necessarily changing the file or its attributes.

Therefore the final dimension for File Integrity Monitoring is to generate an alert when any access to these files or folders is detected, and to provide a full audit trail by account name of who has had access to the data. Much of PCI DSS Requirement 10 is concerned with recording audit trails to allow a forensic analysis of any breach after the event and establish the vector and perpetrator of any attack.



```
192.168.1.221   auth   (warning):  Security 4663: Microsoft- Windows- Security- Auditing: Object: An attempt was made to access an object - An attempt was made to access an object. Subject: Security ID: S- 1- 5- 21- 137452079- 2713887879- 3978310812- 1003 Account Name: mark Account Domain: NNTMKEDGLEY Logon ID: 0x4ecc1 Object: Object Server: Security Object Type: File Object Name: D: \ RESTRICTED ACCESS - Card Reconcilliation Folder Handle ID: 0xd68 Process Information: Process ID: 0xef4 Process Name: C: \ Windows\ explorer. exe Access Request Information: Accesses: % % 4423 Access Mask: 0x80
Details...
```

Figure 3 - File Integrity Monitoring employed to protect secure files from unauthorized access. Whether you are protecting Card Holder Data for PCI DSS compliance or sensitive Financial Information for Sarbanes-Oxley compliance, you will need to log a full audit trail of access to the files and folders concerned. Here we use NNT Log Tracker to illustrate this in practise - note we have also logged the user and process accessing the protected folder.

About NNT

New Net Technologies (NNT) is the leading provider of Secure Ops, which leverages security through System Integrity along with Intelligent Closed Loop Change Control, focused on helping organizations reduce their security risk, increase service availability and achieve continuous compliance. NNT delivers its Secure Ops suite by combining: System Configuration Hardening, Closed Loop Change Control, Vulnerability Management and Event Log Management. These core security disciplines are defined by the SANS Institute as the essential Critical Security Controls for any cyber security initiative.