The Role of FIM in Mainframe Security

How Vanguard's SAMM's tailored tools enhance system integrity monitoring

VANGUARD
INTEGRITY PROFESSIONALS
CYBERSECURITY EXPERTS

WOMEN OWNED

Table of Contents

Executive Summary

1. Why FIM Is Essential for Mainframe Security

2. What's So Special About IBM Z and FIM?

3. FIM Strategies on IBM Z

4. FIM and the Future

Executive Summary

All organizations face the challenge of ensuring cybersecurity and meeting regulatory compliance requirements. One vital component of these efforts is File Integrity Monitoring (FIM). FIM is a method used to verify the integrity of software assets that systems rely on. It ensures that software applications remain exactly as their vendors intended, without any unauthorized changes, across all copies within an organization.

For FIM to be effective, it requires a trusted baseline to compare against. Without this, its benefits are diminished. Despite its importance, many organizations still rely on outdated maintenance procedures that overlook the need to actively monitor and verify the integrity of critical software and data.

FIM plays a crucial role in maintaining the integrity of data and systems on the IBM Z mainframe. While certain IBM Z features have partially addressed this need, there is now a call for more comprehensive insight into system integrity.

A specific example of such a solution for IBM Z is Vanguard Integrity Professionals' Software Application Monitoring and Malware Detection (SAMM). SAMM provides tailored tools and approaches to support FIM and enhance system integrity monitoring on IBM Z.

Key Features of SAMM:

- Allows BASELINEs of systems that can be fetched and locked
- Provides continuous capture of SMF data even if our STC is not running
- Provides an ACCEPT record process where users can mark an activity as acceptable when it as previously flagged as suspicious
- Provides 24/7 file integrity monitoring
- Validates PTF tracking applied across the enterprise
- Alerts on inconsistencies between SMP/E and system datasets
- Categorizes changes as RED (potential malicious behavior), YELLOW (behavior that you may want to investigate), and GREEN (behavior that is expected)
- Delivers continuous cross-system visibility into dataset and module changes, correlating events in SIEM platforms to enable proactive threat detection, rigorous change auditing, and regulatory compliance

Chapter 1

Why FIM Is Essential for Mainframe Security

The spread of hacking and malware in non-mainframe platforms has led to the practice and technology for monitoring the integrity of critical programs and data files to catch and neutralize bad actors and their doings, such as ransomware. While the solidity of the IBM Z mainframe platform has allowed a sense of greater security, it's essential to implement solutions that provide proof that security integrity is maintained.

As the system of record for the world economy, the mainframe is "where the money is," and therefore an ideal target for those who would modify critical programs and mainframe data for illicit gain. At the same time, modifying critical programs and mainframe data may also be a part of normal business operations. FIM is thus a necessary part of a complete security and integrity stack for this system of record, but it must be able to monitor all relevant changes and distinguish which are malicious.

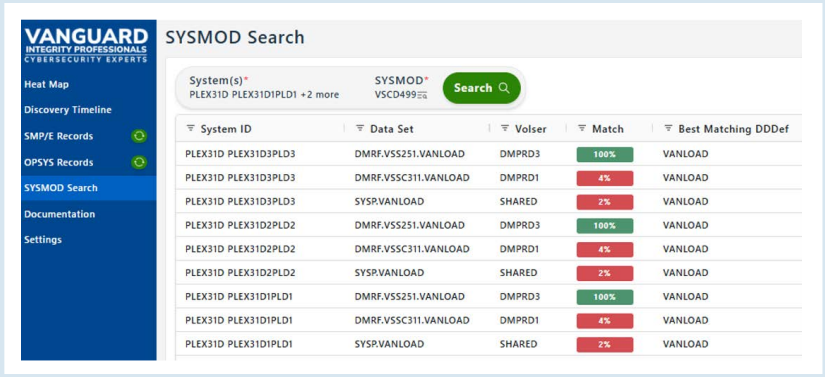
> Talking about real control for your z/OS system datasets, SAMM delivers that by intelligently tracking every critical change and reports on datasets, regardless of encryption status. If a modification doesn't match your verified SMP/E maintenance, SAMM steps in with an immediate alert which allows for users to check an LPAR/system and find out if PTFs applied to the execution libraries are being tracked. Plus, if you ever wished you could instantly check PTF application status across all your systems, no matter where your SMP/E data sets live, SAMM makes that a reality with its powerful search capability. It's about bringing true clarity and proactive security to your mainframe."–*Brian Marshall, Chief Strategist*

FIM Context and Mandates

FIM is about more than just pre-empting and detecting bad actors. Provable integrity has business value, especially to institutions responsible for financial, confidential, or personal data and processing. On top of that is an ever-growing list of business best practices, principles, rules, regulations, and laws in jurisdictions from local to global. Yet, in many ways, FIM began as something of an immunity-type after-the-fact response to a particular set of new threats.

If all critical data and programs could simply be hardened into ROM or absolute volumes, the security and data integrity and processing wouldn't be subject to such a range of exposures. But change is inevitable. Data must increase and be changed: that's what data does. Software must be patched to fix bugs and new security issues and to introduce new functionality. FIM solutions like SAMM are necessary.

SAMM in Action



Entropy

Call it what you will: Murphy's Law, human error, acts of God, fat fingers. Stuff happens–unplanned, unexpected, undesirable, unavoidable stuff. The most resilient systems have availability measured in the nines, but even 99.999999% uptime is not 100%. And there are more dimensions than you can shake a measuring stick at. It makes no sense to assume things will always go as planned. It is essential to monitor and identify when things do go sideways, and to have the ability to respond with resilience and restoration – i.e. FIM.

Malware

Viruses, malicious backdoors, Trojan horses, worms. Pandora's box of self-replicating malware is open, and there's no limit to what will come next. All malware relies on the ability to bypass or modify security, replicate itself, and travel to other environments. In so doing, they may "infect" system configurations and programs. Malicious attacks are now being carried out at high levels of sophistication and span over long periods of time (e.g., XZ Utils backdoor from February 2024). Regardless, if critical programs, configuration, or other data are illicitly modified, catching and responding to that modification with FIM tools is essential.

Ransomware

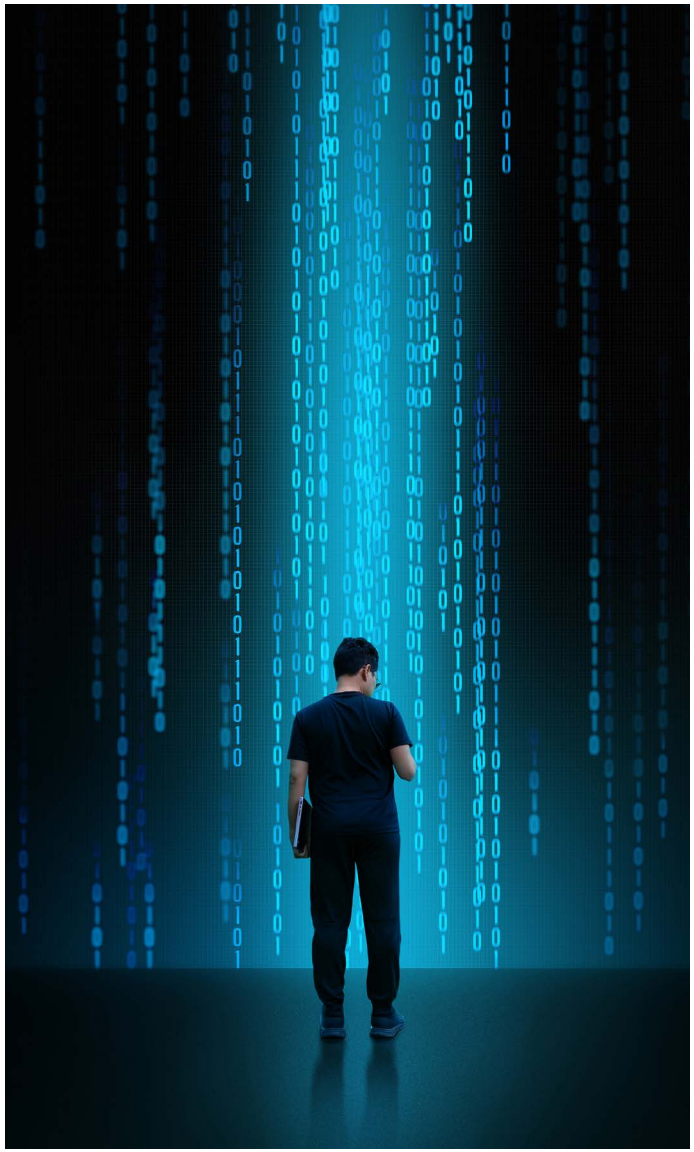
Often following on from malware or arising from similar attacks such as phishing, spear-phishing, social engineering, or any other hybrid or complex approaches, software can be infiltrated to encrypt or exfiltrate critical corporate data, holding that data for ransom. Alerts must be triggered by the modification of critical system configuration and software in order to activate the malware. FIM solutions invoke a response the moment the modifications are underway.

Persistent Threats

Malicious attacks on governments and organizations, that have been detected, are now being carried out gradually over a much longer time. Advanced persistent threats (APT) involve an attacker spending significant time gaining trust and authorization to critical resources while remaining undetected. An APT attack would involve tampering with data or software assets in order to install malware. Often, attackers will take additional time to camouflage their malware, by installing many different pieces that individually do not appear malicious, but act together to create a backdoor. FIM solutions prove software assets are authentic and haven't been tampered with, which is critical for hardening the mainframe platform.

Chapter 2

What's So Special About IBM Z and FIM?



Since it was first available, the IBM Z mainframe has been the system of record for key data and processing in the global economy–a role that appears to remain entrenched for the foreseeable future. While its long-standing presence offers some explanation for its stability, this platform also boasts unique security features. These features have successfully prevented many of the common exposures and incursions seen on other systems.

The IBM z/OS System Integrity Statement, first announced in 1973 for the MVS operating system, is the basis of mainframe security. This guarantee of integrity is backed with a range of features from authentication, authorization, and logging through memory isolation and supervisor state protection. But it only guarantees that the platform is securable, and leaves it up to individual organizations to choose how they wish to properly take advantage of that securability.

While the mainframe is inherently secure, it can still have weak spots. These often come from how precisely and currently it's configured, secured, and updated.

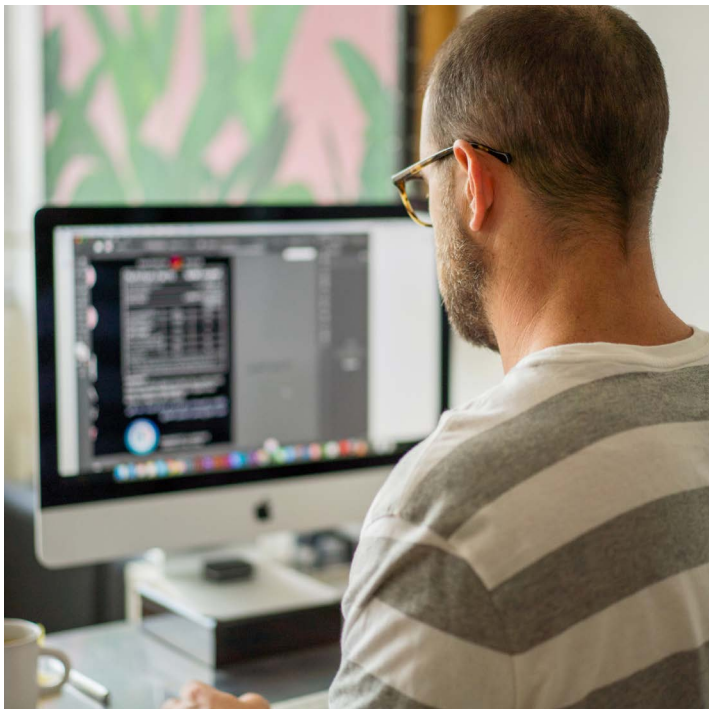
Over time, new security issues that arise from imperfect configuration are commonplace, and need to be proactively dealt with, beginning with using a FIM solution such as SAMM. To clarify further, the following are some of the key data and processing resources on the mainframe which, even if protected using traditional security controls, still need to be actively monitored to avoid corruption that has processing, regulatory, and security implications.

Authorized Program Facility, LPA and LLA

Ironically, one of the most important potential exposures on the mainframe is one of its key security dimensions: the control over the execution of privileged instructions using the Authorized Program Facility (APF).

Application programs and other user functions run in what is known as "problem state" and do not have access to operating system-level functions that could circumvent security measures. Only trusted operating systems and utility software should be allowed to run in the "supervisor state" which enables advanced functionality. The measures that limit access to this advanced state include only allowing programs (or "load modules") that are located in trusted datasets ("load libraries") to perform operating system-type authorized behaviors. This is enforced using secure lists of trusted load libraries.

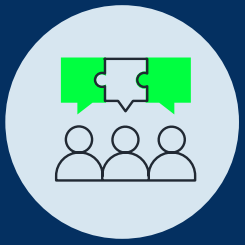
The definitive list of this nature is kept in the system parameter library, (often named "SYS1.PARMLIB") and can be changed real-time using system performance management software and console commands. It's also the case that load modules, which are kept in memory at all times in what is known as the "Link Pack Area" (LPA) are authorized. There are additional load libraries which are defined to the system for fast retrieval of load modules



using the Linked List (LLA), which can be set to have its programs authorized as well, though enabling this feature is not a security best practice.

In all of these cases, both the parameter library and any load libraries containing authorized modules must be strictly secured and monitored. Changes to any of these are significant security exposures if they aren't approved and from a trusted source.

FIM monitors and alerts on all such changes.



What the industry says

A prominent international bank, facing the complexities of regulations like DORA, GDPR, and PCI DSS, recognized a critical need–ensuring every PTF and vital file was installed correctly on their systems. Instead of hoping for the best, they proactively embraced Vanguard's technology designed to validate every single change to critical system areas. This strategic move not only solidified their compliance across global standards but also gave them confidence that their systems were always correctly configured, secure, and monitored.

JCL Libraries

Job Control Language (JCL) is the usual scripting and definition method for most of the tasks that run directly under the control of the mainframe operating system. Text library members that constitute JCL jobs and procedures (PROCs) are often stored in procedure libraries (PROCLIBs), which may be defined directly to the Job Entry Subsystem (JES) or other production or workload automation systems, or located using a PROCnn DD statement in a JCL job.

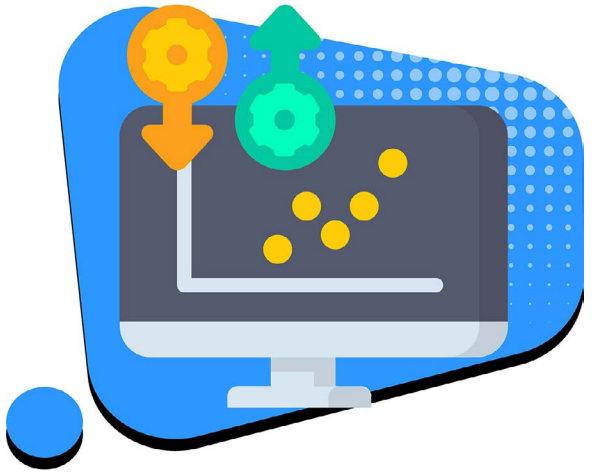
These JCL dataset members are used to run systems and production programs that often modify sensitive data and are consequently sensitive themselves. Any unauthorized modification to them could be a serious security incident. Therefore, it's important to use FIM tools to monitor and alert on any such modifications.

CLIST, REXX, and Other Interpretive Language Program Libraries

Another text-based way that actions are performed on the IBM Z mainframe is using interpretive programming languages, such as CLIST and REXX and a growing list of other such languages like Python.

Changing such programs is as simple as modifying the text library member that contains that program. A rogue modification of such a PDS member could have significant and immediate impacts with a wide range of functional areas, including system automation.

It's therefore essential to be notified of any possibly illicit modifications to such programs immediately, as the consequences are equally immediate. FIM does this.



PARMLIB

In addition to the aforementioned lists of authorized load libraries, the system parameter library (again, often named "SYS1.PARMLIB") contains much of the other configuration information about how the z/OS platform operates. Along with other libraries that contain configuration information for sensitive systems, such information must be tightly controlled and monitored as illicit or erroneous modification can result in significant losses and exposures. These text library datasets often can be updated by multiple user IDs, so tracking the occasions and nature of any updates must be undertaken externally to any one responsible individual.

This is exactly what a FIM does.

STEPLIBs and JOBLIBs

While application and user programs that only run in "problem state" aren't able to invoke operating system functions, they do often have access to critical production data including financial information, personally identifiable information, corporate confidential information, and the whole range of sensitive business information.

The covert modification or replacement of such load modules can have immediate and devastating impacts on an organization's entire business. It's essential to be notified immediately of any suspicious changes to such programs in the libraries where they reside, often referred to in the JCL STEPLIB and JOBLIB statements.

This is an important use of FIM. A FIM solution such as SAMM will provide organizations with an automated method to monitor that the intended runtime libraries are in use and report/alert on any changes.

Source Libraries

While other programming languages run on IBM Z, the powerhouse of the world economy features COBOL running on the mainframe. Adding in the other compiled languages on the mainframe, from PL/I to C to numerous proprietary vendor languages, and all the text-based source versions of these programs amount to an essential resource. Without having a definite version of the source code for each compiled load module in use, it becomes dangerous to try to make any changes and recompile a program.

Source code managers are a normal solution to ensuring that the connection is tracked, but the libraries of such programs (and their add-in portions such as copybooks with standardized lists of variables) make up a significant source of vulnerability to unauthorized modification, managed or not. And given the large number of user IDs that typically have access to them, tracking the source of a rogue modification can become a nightmare when something goes wrong with the compiled load module that should map to that source module.

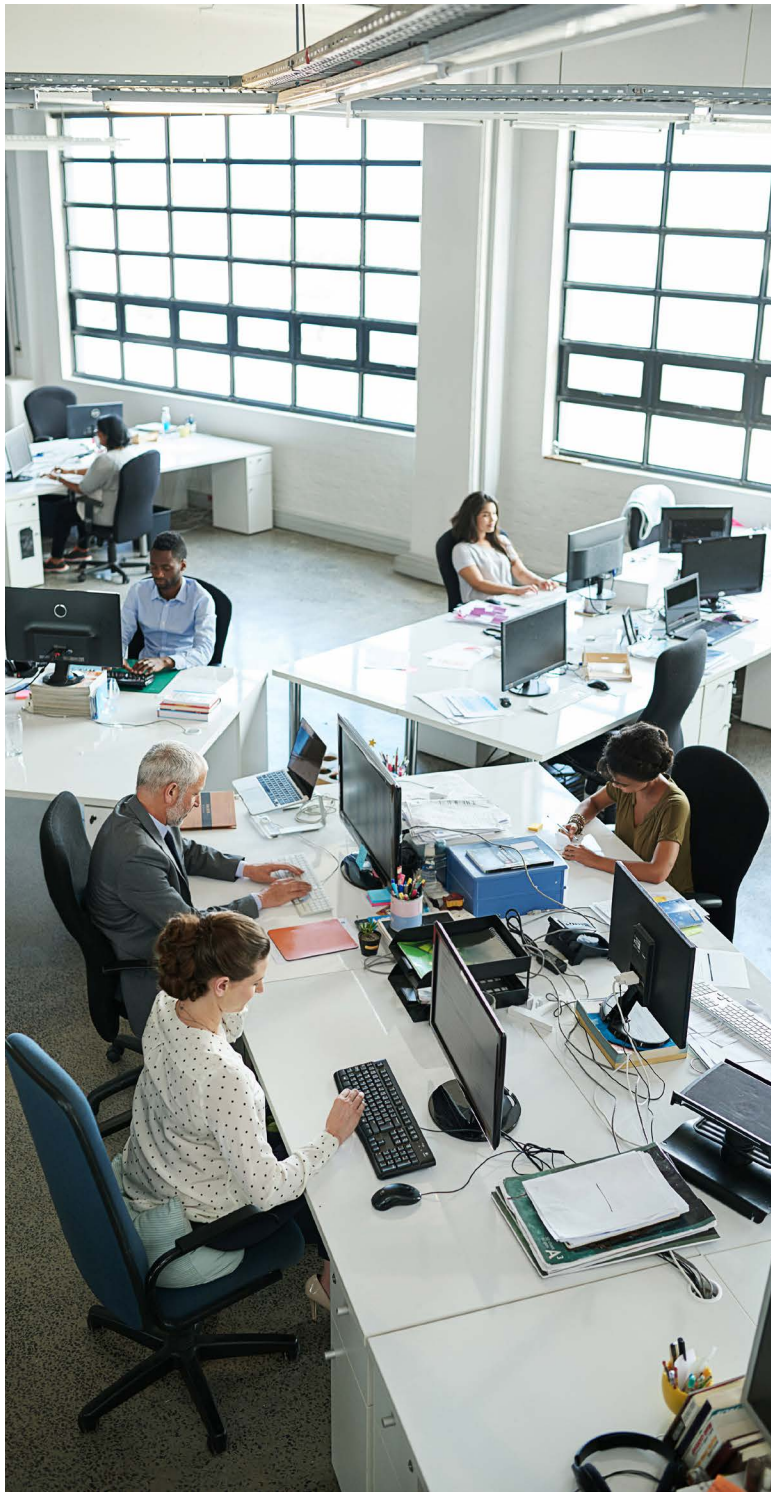
A FIM like SAMM is the missing link to monitoring and tracking such changes and alerting immediately if something inappropriate may have occurred.

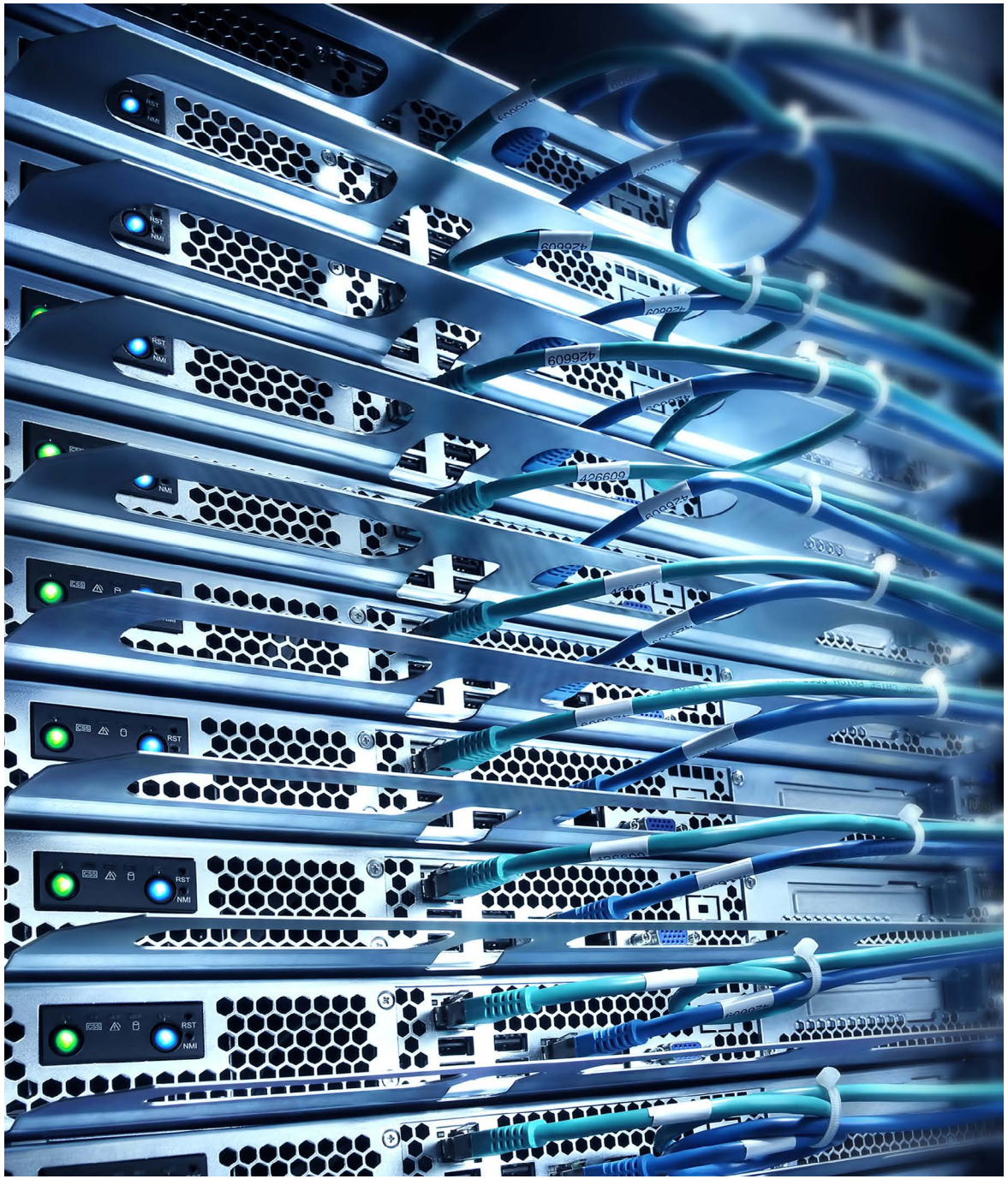
SMP/E-Managed Libraries

Today the distributed term "patch management" is often applied to mainframe maintenance practices, though it's not always appreciated by experienced mainframers. Whether or not the term is accepted, when fixes are distributed to a system, it's important to detect immediately if bad actors have targeted the systems being maintained, whether the products are from IBM or independent software vendors. Also bear in mind that security/integrity fixes are first received into System Modification Program/Extended (SMP/E), where they're then used to modify target and, eventually, distribution libraries. The organization is responsible for ensuring that these fixed versions are placed into all production libraries throughout the enterprise. This makes it vital that SMP/E libraries assets have integrity, because copies of them are likely to be made.

Failure to monitor, track, alert on, and report on changes would significantly dislodge your environment from the safety zone. Authentic maintenance produced by SMP/E must be used by all production libraries. Any maintenance applied to a load module must be accurately detected and recorded. At the same time, the system should avoid false positives if the module's contents are merely reordered without any actual additions, deletions or functional changes.

As a proper mainframe FIM solution, SAMM can distinguish between genuine and unwelcome modifications across all production and SMP/E libraries independent of a mere reordering of constituent sub-modules. It also can independently verify that members in target libraries match exactly what is produced by the applied PTF or APAR.





Data Stores and Databases

Why would someone commit a ransomware attack on a production environment such as the IBM Z mainframe? Because that's where the data is. Critical mainframe data sources such as proprietary tables are essential to the applications that use them.

When a normally static reference data source is unexpectedly modified, this must be immediately detected and then raise an alert if the data is in any way critical. And that's what a FIM such as SAMM can do.

Application and Third-Party Non-SMP/E Load Libraries

Not every application or product in use in a mainframe environment uses standard installation and maintenance methods such as SMP/E. Regardless, if an organization deploys programs, load libraries, or configurations that could be modified by someone who's misusing authorization, the implications for your enterprise's integrity demands a FIM solution to respond immediately.

Chapter 3

FIM Strategies on IBM Z

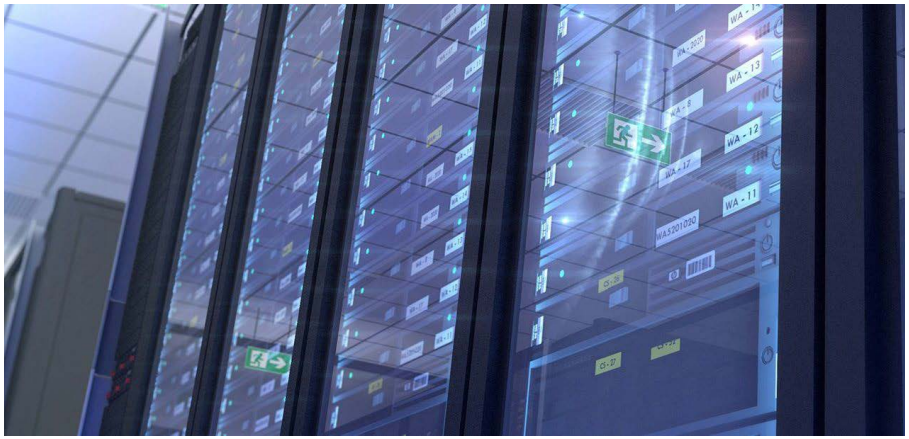
Establishing a fully functional FIM solution, such as SAMM, in this environment entails many factors and strategic approaches.

FIM Baselines and Integrity

Two central aspects of a FIM solution are:

1. Having a clear and independent, authoritative baseline picture of what the relevant data and programs in an uncorrupted system look like.

2. Having the ability to reliably detect what has been changed and when, in order to compare the result to the baseline.

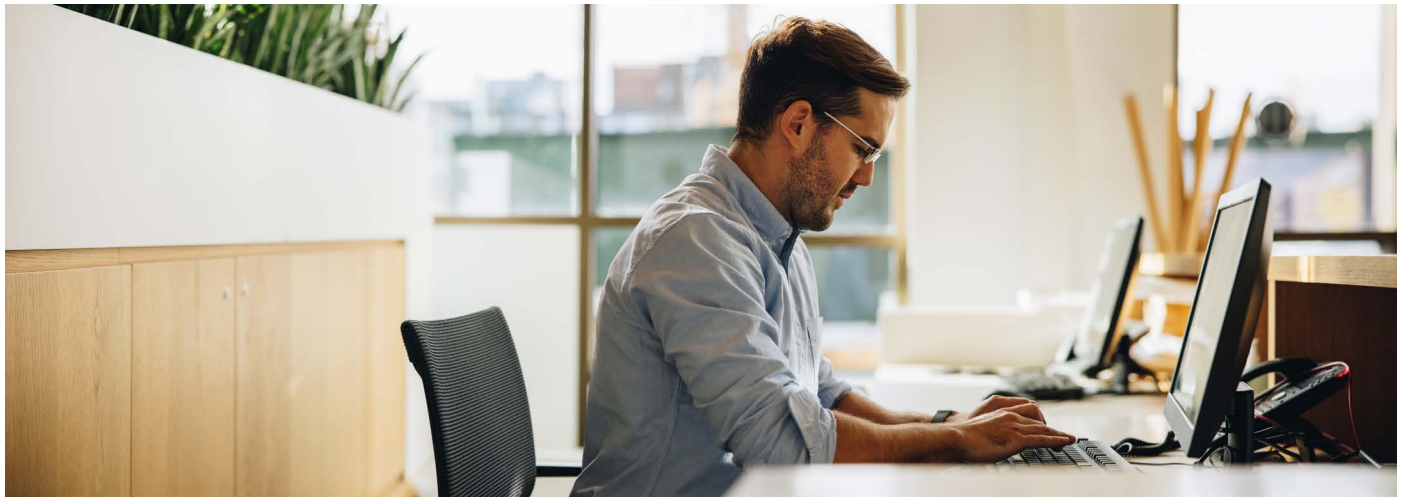


It's necessary to do frequent and efficient compares of changes soon after they are made to avoid serious damage. And it's important to minimize false positives by recognizing when a change is insignificant, even if it reorders the sub-modules in a load module.

Based on these requirements, especially if the number of critical datasets and their total size are very large, an efficient and fully reliable formula is preferable over an exact mirror of the data in question. A method such as a signature or checksum using a recognized hashing algorithm can save on both baseline size and comparison time. A rapid hashing of the changed data and comparison of signatures is much faster than a linear byte-by-byte comparison.

When considering the maintenance of executable load modules, it's essential that it propagates out to all libraries. A FIM solution must implement a reliable hashing method so it recognizes a load module is unmodified, even if constituent elements have been re-ordered. An efficient FIM solution would be capable of searching where a PTF or APAR presence is detected and where it's absent across all libraries. This would equip organizations with a tool that can reliably track the progress of deploying maintenance to production systems, especially when the maintenance involves security/integrity PTFs and APARs.

For these reasons, an efficient formula approach, like that taken by SAMM, explicitly takes these matters into account.



Choosing Targets Worth Monitoring

When IBM introduced pervasive encryption to the Z mainframe, allowing any data to be stored encrypted and dynamically decrypted only when needed for use, it quickly became clear that many objects weren't worth the resource costs of constant encryption and decryption. System files such as the programs that run the platform, for example, weren't an efficient choice.

Not every load module or data source on the platform is sufficiently critical to receive the attention that a FIM solution can bring. But the criteria are somewhat different, because the implications of modifying critical programs and data are different from keeping them confidential–and, in fact, for some critical system programs, it's a preferable option to find out if they've been modified than to constantly decrypt them.

In both scenarios, a proper inventory of critical load libraries and data sources should be made, and recommendations and decisions should be carefully carried out. SMP/E and similar maintenance environments, especially for products that touch consequential data or employ APF-authorized programs, are examples of targets worth monitoring.

Securing Critical Data With SAMM

FIM is here for the IBM Z mainframe, and not a second too early. In the three-plus decades since FIM was envisioned for internet-connected computers, the IBM Z platform has become extremely integrated with the internet in many ways. Hence, relying exclusively on traditional security is no longer enough.

As part of the ongoing journey of maintaining the integrity of the system of record for the world economy, a FIM like SAMM is now a required aspect of a full security program, ensuring that the critical data and processing aren't only protected but also have their integrity proactively guarded and monitored.

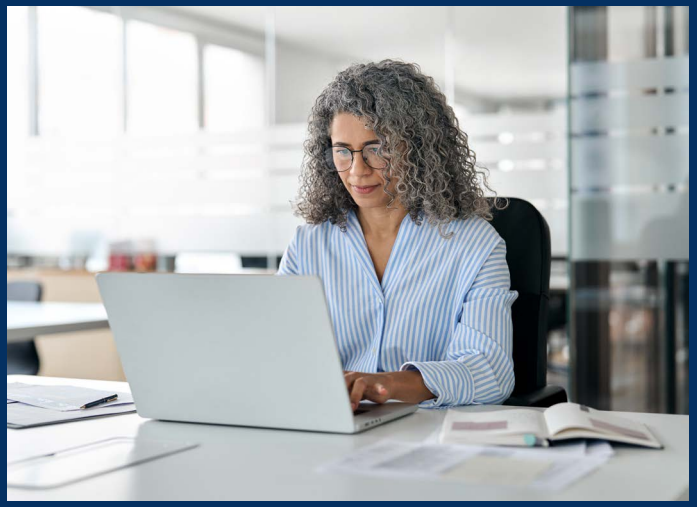
Mainframe Security Monitoring Mechanisms

IBM's Z mainframe platform offers a wealth of ways to monitor changes, and the monarch of these would be the System Management Facilities (SMF). Should you choose to enable the creation of SMF records every time a dataset is changed, a strong FIM product can examine each event in real-time to determine if an alert is necessary. However, this volume of real-time data monitoring incurs a certain amount of overhead.

Some sites avoid the creation of large volumes of SMF data, preferring to instead do regular checks on critical datasets to see if anything has changed, and investigate further when a modification has been uncovered. In this case, a trade-off exists between high frequency verification, which uses more resources but can more rapidly detect and respond to illicit changes, and lower frequency, which is gentler on resource usage but may leave the window open longer, potentially enabling bad actors to get a foothold.

There is also the issue with direct access storage devices shared across systems. A FIM product on the mainframe must be able to detect changes to load modules and text dataset from other systems without relying on the SMF data on the system being monitored.

On top of platform-local considerations, there's also the matter of sending alerts and heartbeats to an enterprise SIEM for security operations awareness, monitoring, and sophisticated correlation and alerting. Hence a FIM must offer these features.



What Customers Want in a FIM

Peace of mind and proof of integrity is likely the most honest answer a customer of a FIM solution would give when asked why such a thing matters to them. The moment it becomes clear that they don't have the resources to ensure that all of their critical programs and data are secured from any type of intentional or accidental corruption, the need for an additional dimension of awareness to catch any violations of such integrity becomes apparent.

Of course, a solution that already exists, is easy to install, and configure and use, and runs with low resource usage and high reliability is essential.

It's fortunate that FIM solutions such as SAMM now exist for the IBM Z mainframe platform, as they're no longer optional in order to meet the ever-increasing compliance requirements that have well-defined penalties.

Chapter 4

FIM and the Future

FIM is about preserving the integrity of systems of record, including the IBM Z mainframe platform. The history of business computing to this point in history has made it clear that we often cannot predict future threats until we've experienced them, much like the human body's immune system response.

However, whatever future attacks look like, they'll either modify critical data or processing, or take advantage of emergent weaknesses that must consequently be patched. In all of these cases, and in all the established matters of maintaining integrity for the systems of record, a FIM such as SAMM is now a permanent necessity. The future of FIM solutions must involve increased reliability in recognizing load modules, integration with maintenance operations and procedures, and efficient use of resources.

Future Regulations and Other Factors

On top of emergent security exposures, there will be unlimited additional regulations worldwide, all of them in some way relevant to the sensitivity of data and processing. FIM solutions will continue to be critical as they evolve to respond to the future requirements that are still developing today.

What does the market say?

Facing strict compliance demands like 23 NYCRR 500, a major New York regional bank sought an aggressive solution. They found their answer in Vanguard's SAMM, a technology they now highly recommend for its effectiveness in meeting complex regulatory requirements for both internal and external audit requirements.

VANGUARD
INTEGRITY PROFESSIONALS
CYBERSECURITY EXPERTS

WOMEN OWNED

Connect with us to see why SAMM is the FIM you need.

Go2Vanguard.com

702-794-0014

Biz.Dev@Go2Vanguard.com

About Vanguard Integrity Professionals

Founded in 1986 to help customers safeguard mission critical applications and data, Vanguard Integrity Professionals is the largest independent provider of enterprise security software for addressing complex security and regulatory compliance challenges. Vanguard continuously drives innovation in security software and technology to stay ahead of evolving regulatory requirements in an ever-changing threatscape. Led by some of the most knowledgeable minds in the cybersecurity industry, our security solutions take the lead.