

Host Integrity Monitoring: Best Practices for Deployment

Brian Wotring 2004-03-31

Introduction

There are now a number of commercial and open source solutions that can be used to effectively monitor the integrity of host environments. When understood and used correctly, these applications can be very helpful with detecting unauthorized change, conducting damage assessment, and preventing future attacks. With all that is involved in deploying such a system, there are some very important concepts that are often neglected, not understood, or not expressed in the documentation.

The purpose of this article is to highlight the important steps and concepts involved in deploying a host integrity monitoring system. Being aware of these concepts can mean the difference between a useful deployment, and one that is rendered ineffective or more trouble than it is worth.

This article is written with the open source host integrity applications Osiris and Samhain in mind, however the material presented is certainly not unique to these applications.

1. Understand the big picture

Network-based monitoring tools garner a lot of attention because they provide packet-level visibility into events that affect multiple machines. However seeing the packet sent by an attacker to a vulnerable host only warns you that something has happened, usually when it's too late. In order to identify how the host has responded and whether or not the attack was successful, you usually have to look at the target system. Host-based monitoring tools give granularity that makes attacks visible on the host on which they are installed.

There are many ways of establishing visibility at the host level. However, it is important remember that no system or application by itself can be trusted with the task of providing assurance of host integrity. In fact it is best to have more than one way of detecting suspicious behavior or unauthorized change. There are many analogies for this. One that is easy to draw because of its commonality in our society is that of a bank. Banks do more than just lock their front door. They often have cameras, motion detectors, armed guards, and locked vaults. An attacker may go unnoticed by one of these monitoring measures, but is not likely to get past all

of them. This concept of protecting at multiple layers is called defense in depth.

It is important to understand that deploying a host integrity monitoring system such as Osiris or Samhain is part of a larger effort to provide visibility at the host level. These applications are the most effective when combined with other practices such as proper log configuration and analysis, remote logging, and keeping your hosts up to date with security patches.

The basic idea behind host integrity monitoring applications is that they detect and report on change to the system. It gets most interesting when a change is unauthorized or unwanted. Much of the monitoring is focused on the file system. However, other environmental vectors can be monitored as well. For example, Samhain has the ability to search for rootkits and monitor login and logout activities. Osiris has the ability to monitor the state of loaded kernel extensions and the details of changes to the local user and group databases. Detected change is reported in the form of log files, syslog, the Windows Event Viewer, and possibly emailed to an administrator.

Some of the situations where knowledge of changes to a host is essential include, but aren't limited to: servers running common services (mail, www, DNS, CVS, NFS) that are exposed to the Internet, production build environments, Certificate Authority and key servers, firewalls, hosts used for testing that require a consistent environment, and database servers.

To better appreciate the role a host integrity system serves, imagine you find a new link to `/etc/passwd` that has been created in `/tmp`, a new kernel module that gets loaded without your knowledge, or a new user gets mysteriously created. How would you know if and when these types of changes occurred? There are commands that can be used to look for these happenings, but how would you know if and when to run them? What if these commands that you depend on for finding such changes were altered to hide specific information? Now, imagine you have hundreds of hosts that need to be monitored regularly to look for changes such as these.

The main point here is that host integrity monitoring software can serve a significant and distinct role in your security policy. Since such programs are implemented as software, they are not perfect and are also at risk for compromise. However, when other countermeasures are in place to protect the system and proper logging is in place, host integrity monitoring software serves as yet another hurdle for an attacker to defeat and can provide the first indication of a break-in or compromised host. When properly configured and deployed, this type of software is a powerful addition to the layers that defend your infrastructure in depth.

2. Prior planning prevents poor performance

It cannot be emphasized enough how important planning is before deploying software of this nature. Every application has different strengths and weaknesses that should be weighed against your goals or requirements. This section lists some practical questions that can aid in the planning process and a table that contrasts the features of common open source host integrity tools.

How many hosts are to be monitored?

If you only need to monitor a single host, a system such as Osiris is probably not the best choice because it is engineered for managing multiple hosts. However, Samhain is very easy to install and configure to monitor a single host (Samhain can also be used to monitor multiple hosts).

This question also plays a role in how your management console will be implemented -- more on that in the next section.

How is your network organized?

It is not uncommon for there to be monitored hosts spread across different networks, or a DMZ. It may be ideal to deploy multiple consoles, one per network. This is generally a better practice than accommodating a single management console by punching holes in your firewall rules. Secured sections of the network are usually segregated for a reason and the deployment plan should respect that reasoning.

What operating systems and versions are on your list of hosts that need to be monitored?

Most open source host integrity monitoring systems available today are targeted for UNIX and Linux systems. Osiris supports the NT based Windows operating systems (NT,2K,XP) and Mac OS X, Linux, FreeBSD, OpenBSD, IRIX, AIX, and Solaris. Samhain does not run on Windows, but does support UNIX based systems such as, AIX, HP-UX, Unixware, and Alpha/Tru64 UNIX.

It is very important to pay attention to what platforms and versions the vendor or project is claiming to officially support -especially for open source products. Just because an application runs or compiles does not mean that it is safe to deploy it in a production environment. These

tools often deal with platform specific elements of the system and many of the features may not work as expected. Moreover, it may be more risky to your network to deploy an untested application of this nature than to do nothing at all.

Who will manage and administer the system?

The easiest scenario is a single administrator. However, authority for hosts is often spread across multiple administrators. Osiris and Samhain can both be centrally managed. Osiris has the ability to create user accounts for each administrator and activities are logged on a per-user basis. Access control isn't granular enough to scope an individual user's access to only the resources they are responsible for. Samhain makes use of a centralized logging facility called, "yule". Each host can be directed to a separate log file and access control can then be handled via file permissions.

What elements of the systems will need to be monitored?

This is one of the more difficult questions to answer as each environment has its own unique set of challenges. The biggest distinction to be made is probably between server and desktop environments. Servers generally have a collection of critical files that should not change very often. The basic idea is that these critical elements should be monitored frequently for change. The growing problem with desktops is that they are targets for viruses and worms. Although the host integrity applications listed below can be used to monitor desktops, they are generally more effective in the server environment.

All of these systems allow for the specification of what files or environmental elements should be included in the scans -some with more granularity than others. One thing to keep in mind is how files will change. For example, log files (by nature) change in content and size on a regular basis. It is safe to ignore these changes. If the permissions, owners, or inodes of log files changed, one might wish to investigate, assuming that it wasn't time for the logs to be rotated. Both Osiris and Samhain ship with default scan configuration files for common platforms. These defaults are reasonable to start out with, however you should customize them to accommodate your system(s).

Table One, below, compares features of some popular host integrity applications.

	Samhain	Osiris	INTEGRIT	AIDE
--	----------------	---------------	-----------------	-------------

Monitors Files	yes	yes	yes	yes
Monitors Kernel	yes	yes	no	no
Platforms	Linux, FreeBSD, AIX 4.x, HP-UX 10.20, Unixware 7.1.0, Solaris 2.6, 2.8, and Alpha/True64	Windows NT/2k/XP, Mac OS X, Linux, Solaris, FreeBSD, OpenBSD	Linux, FreeBSD, Solaris, HP-UX, Cygwin	Linux, FreeBSD, OpenBSD, AIX Unixware 7.1.0, Solaris True64, BSDi, Cygwin
Multiple Administrators	no	yes	no	no
Supports Modules	no	yes	no	no
License	GPL	BSD style	GPL	GPL
Centralized Management	yes	yes	no	no
Signed Databases	yes	no	no	no
Database Integration	yes	no	no	no

Table One: a comparison of popular host integrity applications

More information on the above products can be found on their websites:

Samhain - <http://la-samhna.de/samhain/>

Osiris - <http://osiris.shmoo.com>

INTEGRIT - <http://integrit.sourceforge.net/>

AIDE - <http://www.cs.tut.fi/~rammer/aide.html>

3. The management host: cornerstone of the deployment

Establishing trust for any host integrity solution begins with trust of the management console.

The reliability of information stored on and manipulated by hosts that are managed by the management host is directly related (but not limited) to the integrity of the management host. Deployment of the management host should be handled in a similar fashion to the way that one would deploy a log server; after all, they play similar roles. Aside from general security concerns with locking down any host, the following are some considerations specific to deploying a host integrity management console:

Use a Dedicated Host.

If at all possible, dedicate a host or set of hosts to managing the information collected by the scan agents. Ideally these hosts will only perform this function. The more services that are run on a host, the greater the potential for that host to be compromised. More services can necessitate more administrators, more user accounts, more changes to the system, more open ports, and more software to keep patched.

Restrict Access As Much As Possible.

Access to the management host should be restricted as much as possible. This includes both physical and network access. Ideally, the management console will be located in an area where physical access can be limited to the administrators and logins are only possible from the console.

In an idyllic world, the only open ports on the management console will be those used by the host integrity system itself. Obviously, this will be easier to do if the host is used strictly as a management console as described in the previous section. At a minimum, the management host should firewall itself. Putting it behind a dedicated firewall is even better.

Finally, only create user accounts for the administrator or administrators.

Monitor the Management Host.

The first host that should be added to the list of managed hosts is the management host itself. An alternate strategy is to create multiple management consoles that are setup to monitor each other.

4. Baseline integrity

Most host integrity applications work by detecting differences between the current environment and a trusted data set. This trusted data set is often referred to as the "baseline".

Ideally, a host integrity monitoring agent is installed on a host before it is ever deployed or placed onto a network. The monitoring software and its host operating system should be verified as described in the next section to ensure that the generated baseline can be trusted. There are ways of doing this with each of the systems listed above, but the process is not always an easy one.

Although creating baselines before deployment is ideal, it is not always practical. Installing a host integrity monitoring system after host deployment is not perfect, but that doesn't mean it should not be done. If you are unsure of the current state of the host to be monitored, it behooves you to give the host to be monitored some tender loving care. If you install the monitoring software on a host that is compromised, it will make a baseline of that host. Since the baseline is of a perturbed state, the monitoring software won't warn you of the compromise.

Once baselines have been established, the first thing that should be done is to archive them onto read-only media and lock them away in a safe place.

5. Building and deployment

Verify software against a known good.

Whether you are compiling from source or using a packaged binary distribution of some sort, it is essential that the files be verified. Often software distributors will make available cryptographic checksums of the file(s). Some take it to the next level and sign them. The important thing to remember is that the verification process must be done in a trusted environment. If the software needs to be compiled, this may involve downloading and burning to read-only media, then performing the verification and build process offline in a trusted build environment.

Note: if building from source, remember to pay careful attention to the platforms the product has been tested on. Give attention to the configuration and compilation warnings; some features or safety measures may end up being disabled despite successful compilation. Since the algorithms that monitoring software use are generally cryptographic in nature, it is critical

that the monitoring software be installed on hosts that have good sources of randomness.

Running Agents from Read-only Media.

If a host is compromised, it is possible for an attacker to spoof the scans regardless of whether or not the baseline was generated prior deployment or not. This is a problem because the agents are nothing more than software themselves.

One way to deal with this is to run the scan agents from read-only media such as a floppy disk, CD, or DVD. This is always a good idea because it prevents the software from being modified by an attacker or for any other reason. This may be easy to do for a handful of critical servers, but for hundreds of hosts it may not be practical.

Another thing to consider with read-only media is where the scan data is stored. Some host integrity applications store the trusted data set on the monitored host and update that same database when the changes are approved. In this case, it will be necessary to go through the process of creating an updated version of the read-only data with each detected change. If a floppy disk is used, this can be accomplished by toggling the write protect. Given the amount of data that can be generated on current systems, a floppy disk may not have enough space to store the executables and data set.

Some systems (such as Osiris) never store scan data on the scanned host, instead it is always stored on the management host. Samhain offers the ability to store the data on the management host and have it pushed to the client only for comparison to minimize the window of opportunity for modification. With these scenarios, the scan agent software is the only thing that needs to be stored on read-only media.

If the scan agent software is not stored on read-only media, both Osiris and Samhain sport features to mitigate the risk of tampering. Samhain offers a handful of self-integrity checks including signed data and configuration files, a key compiled into the executable, and the ability to hide itself from the system process list. Osiris makes use of a runtime session key that acts as a means to authenticate to the management host as well as to detect tampering. Both Osiris and Samhain run as a daemon and therefore the daemon process can be monitored for start/stop/restart events.

6. Logs are important and you should read them

One of the easiest ways to render a host integrity monitoring system useless is to ignore the logs. One of the most important aspects of setting up the management host is to ensure that critical log messages are received, detected, propagated to the appropriate administrator, and are eventually archived.

Use a Log Analysis Application.

Host and file integrity applications are notorious for creating noisy reports and bothersome notifications. Log analyzers are useful because they can reduce noise and many are capable of performing some sort of action (such as email or paging) based upon the detection of a specific log entry.

Using a log analysis application to keep tabs on the overall state of the management host is recommended. It is also important to use that same application to monitor the output of the host integrity application. Finally, using a separate log analysis application as opposed to using built-in analysis and notification mechanisms makes it easy to manage the logs of multiple host integrity monitoring applications.

A generous listing of log analysis applications and related information can be found on <http://www.loganalysis.org> run by Tina Bird and Marcus Ranum .

Understand Log Signatures.

Most host integrity monitoring applications are capable of reporting changes to the Event Viewer on Windows or to Syslog on UNIX and UNIX like hosts. With Osiris and Samhain, for example, there are specific log entries that should be monitored. Usually there are codes associated with specific events to facilitate analysis. Review these codes and make the necessary changes to the configuration of the log analysis application.

Log and Archive As Much As Possible.

It is not always easy to determine what information will be useful in a forensic situation, or in analyzing suspicious behavior in the future. So, the more information you can gather and archive, the more likely you'll be able to reconstruct events that occurred during a compromise.

Perform Regular Backups.

This includes logs, databases, and configuration files. Creating regular archives of these sets of data on read-only media is always a good practice. This is especially true of the baseline scans for monitored hosts. It is important that the management console not be the only location where all of this information is kept!

Conclusion

This article has highlighted some of the most important things to consider when deploying a host integrity monitoring solution. Often these points are left out or not stressed enough in documentation. Always keep in mind your overall goal. Plan ahead. Most of the applications listed in this article are very different in their approach to monitoring a host environment and some will be better suited for your needs than others. Guard the management host with your life, or preferably, your manager's life. Verify software before installing and use read-only media if possible. Last, but not least, read and analyze the logs.

About the author

[Brian Wotring](#) is the lead developer for the Osiris project and CTO of Host Integrity, Inc. Brian is the co-author of "Mac OS X Security" and a long-standing member of [The Shmoo Group](#), a non-profit think tank comprised of security and cryptography professionals. For questions or comments on this article, send mail to: brian at shmoo dot com. I would like to thank Tina Bird, Rodney Thayer, Holt Sorenson, Scott Hallock, and other members of The Shmoo Group for helping me with this article.

[Privacy Statement](#)

Copyright 2006, SecurityFocus