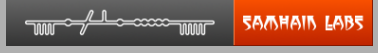

The Samhain HIDS

Overview of available features

Rainer Wichmann

November 1, 2011

Supported Platforms

- POSIX (e.g. Linux, *BSD, Solaris 2.x, AIX 5.x, HP-UX 11, and Mac OS X.
- Windows 2000 / WindowsXP with POSIX emulation (e.g. Cygwin).

  Please note that this is tested for the Samhain monitoring agent only, not for the server.

PCI DSS Compliance

The Payment Card Industry (PCI) Data Security Standard (DSS) mandates the use of file integrity monitoring software. Version 1.1 of the PCI DSS includes the following two requirements:

- Sect. 11.5: *"Deploy file integrity monitoring software to alert personnel to unauthorized modification of critical system or content files; and configure the software to perform critical file comparisons at least weekly."*

- Sect. 10.5.5: *"Use file integrity monitoring and change detection software on logs to ensure that existing log data cannot be changed without generating alerts (although new data being added should not cause an alert)."*

While weekly checks can be done with any open-source file integrity checker, to the best of our knowledge Samhain is the only one that can perform incremental checks on growing logfiles (i.e. verify at each check that the data present at the preceding check have not been modified meanwhile), as required by Sect. 10.5.5 of the PCI DSS.

Centralized Management

Samhain can be used standalone on a single host, but its particular strength is *centralized monitoring and management*. The complete management of a samhain system can be done from one central location. To this end, several components are required. A full samhain client/server system is built of the following components:

- The samhain file/host integrity checker
  - This is the client/agent on the monitored host(s). It is designed to run as a daemon. This avoids repetitive warnings, because the daemon keeps a memory of file changes. However, if you prefer, it can also be invoked from `cron`.
- The yule log server
  - Yule collects and logs reports from samhain clients on remote (or the local) hosts.
  - Yule allows Samhain clients to download *baseline databases* and *configuration files* at startup.
  - Yule keeps track of the status of clients, and can inform you if a client seems to be dead.
  - Yule can advise the client to reload the *runtime configuration file* after an edit.
- A relational database
  - The server will store reports from clients in this database. Oracle, MySql, or PostgreSQL are supported. While a database is in general not required, it is neccessary for the next component.
- The Beltane web-based console
  - Beltane is a PHP application available as separate package.
  - Beltane will pull reports from the database, and present them for review.
  - Beltane allows the user to update the client's baseline databases stored on the server, to reflect the file system changes reported by the client.
  - Beltane II (the commercial version) offers significantly enhanced performance and many additional features.
- The deployment system
  - This is an optional component to facilitate deployment of samhain clients. If your local setup allows to ssh as user 'root' to client machines, the deployment system (which is part of the samhain distribution) provides for simplified *mass deployment* of clients: complete installation of a client can be done with just one command.

File Integrity Checks

- **Immediate notifications, reduced I/O load** On Linux, samhain 3.0+ can leverage the *inotify* mechanism to monitor file system events. This allows to receive immediate notifications about changes, and eliminates the need for frequent file system scans which may cause a high I/O load.
- Checksum (TIGER192, SHA-1, or MD5), size, mode/permission, owner, group, creation/modification/access time, inode, number of hardlinks, major/minor device number (devices only), and linked path (symbolic links only) can be checked.
- Samhain can also check more 'exotic' properties like: SELinux attributes (on Linux), POSIX ACLs (on systems supporting them), Linux ext2 file attributes (as set by chattr, e.g. the immutable flag), and the BSD file flags.
- Who did it? On Linux, samhain can leverage the Linux kernel audit system to determine which login user modified a file. (Please note that in general, i.e. on arbitrary POSIX operating systems, this is not possible to do. The required information is usually not saved by the operating system.)
- Correct number of hardlinks for directories can be checked (this can reveal the presence of subdirectories hidden by kernel rootkits).
- For small files (less than 9200 bytes after zlib compression), it is optionally possible to store the full file in the baseline database, such that it is possible to find out what has been changed.
- Twelve different policies (i.e. different subsets of file properties to check) are available. Each can be fully re-defined by the user.
- The recursion depth (level of subdirectories) can be set globally, or individually for each specified directory.
- Shell wildcard patterns (globbing) can be used to specify files and/or directories to check.
- Exclusion of individual subdirectories and/or files is possible.
- With the 'userdir' extension compiled in, paths can be specified relative to the home directories of all users within some range of UIDs (i.e. it's possible to say 'check /.profile for UIDs in the range N to M').
- On Linux, prelink can be supported transparently (i.e. no spurious warnings after re-prelinking).

Scheduling of File Checks

- File checks can be performed at user-defined intervals.
- Alternatively, a crontab-style schedule can be used to perform file checks at user-defined times.
- It is possible to configure two different schedules, to check some files or directories more frequently then others.
- File checks can be started anytime by sending a signal to the samhain daemon.

Host Integrity Monitoring

Samhain is extensible by modules that can be compiled in at the users' discretion. The following list shows which modules are currently available. The documentation contains a HOWTO for creating additional modules.

Logfile monitoring/analysis

- As of version 2.5.0, samhain optionally perform logfile monitoring/analysis. Currently supported formats are UNIX syslog, Apache (and compatible) access and error logs, Samba logfiles, and BSD-style process accounting logs.
- Both whitelisting and blacklisting policies are supported. Perl-style regular expression can be used to match logfile entries.
- Samhain supports checking for correlated events, for missing heartbeat messages, and automatic detection of bursts of repeated messages.

Windows registry check

- On Windows/Cygwin, it is possible to check the integrity of individual keys (or hierarchies of keys) in the registry.

Kernel integrity

- On Linux and FreeBSD/OpenBSD, samhain can optionally check the integrity of the running kernel to detect kernel rootkits.

SUID/SGID files

- Samhain can optionally check the filesystem for new SUID/SGID files.
- This check can be scheduled independently from the regular file check.
- Optionally, samhain can strip SUID/SGID permissions from new SUID/SGID files, or quarantine or delete them. By default, it will just report them.

Open ports

- Samhain can optionally monitor which ports are open on the local host, and compare agains a list of allowed or required port/services.
- On Linux and FreeBSD, also the program having the port open will be reported.

Host Integrity Monitoring (cont.)

Process check

- Samhain can optionally check for processes that are hidden (i.e. not listed in the output from 'ps'), or fake (i.e. listed by 'ps', but non-existent). Additionally, it is possible to check for the existence of (user-defined) required processes.

Mount check

- Samhain can optionally monitor the presence and mount options of mounted filesystems.

Login/logoff events

- Samhain can optionally monitor and report login/logoff events.
- This check uses the systems utmp file.

Log Facilities

The verbosity and on/off status of each log facility can be configured individually.

- Central log server. Messages are sent via encrypted TCP connections. Clients need to authenticate to the server.
- Syslog.
- Console (if daemon) / stderr.
- Log file. To prevent unauthorized modifications of existing log records, the log file entries are signed.
- E-mail (built-in mailer). E-mail reports are signed to prevent tampering. It is possible to configure different filters for different recipients.
- Database (currently MySQL, PostgreSQL, and Oracle are supported; support for unixODBC is untested).
- Execute external program - this can be used to implement arbitrary additional logging facilities, or to perform active response to events.

Integration with other Systems / Active Response

Prelude

- Samhain can be compiled against the libprelude library, which will enable it to function as a prelude sensor.

Nagios

- A Perl plugin for Nagios (check_samhain.pl) is supplied as part of the samhain distribution.

Generic interfaces

Samhain offers several generic interfaces to communicate with other processes:

- Named pipe: samhain can write log messages to a named pipe.
- Message queue: log messages can be provided on a message queue (SystemV IPC)
- External programs can be executed from samhain. The log message is supplied on standard input.

Active response

- Samhain can execute external programs upon user-defined events, and supply the corresponding log message to them. This can be used to:
  - implement additional logging facilities, or
  - provide active response (e.g. rebooting the machine, reconfiguring the firewall, ...).

Integrity of the Samhain System

There is always a trade-off between security and convenience, and thus you may want to keep your file checking executable on disk and hope that an intruder will not tamper with it. Samhain offers the following features to help protecting its integrity:

- *Signed database and configuration file*: both the file signature database and the configuration file can be signed with GnuPG.
- *Embedded password*: for successful connection to the server, a password is required which is embedded into the executable itself (this password can be set exactly once after compiling the executable).
- *Compiled-in key*: every executable built from source contains a unique random 64-bit key (unless this key is defined by the user at compile time). Logfile/e-mail messages generated by a different executable (with a different key) will not pass the verification routine with a 'known good' executable.
- *Daemon mode*: samhain can run continuously as a daemon (background process), and any stop/restart process will leave a recognizable mark. Thus it is not possible to "slip in" a rogue executable as long as the daemon is running.
- *Signed reports*: e-mail reports and messages written to the log file are signed. Messages sent to a central log server are additionally encrypted with Rijndael (AES).
- *Stealth*: what an intruder cannot see, s/he will not try to subvert. samhain offers several options to hide itself rather efficiently (for Linux, there is even a hiding kernel module available).

Documentation

- Both online and in the `docs/` subdirectory of the distribution tarball, you will find a detailed manual in PS and HTML format, including (but not limited to) explanation and examples for the setup, complete documentation of the format of the configuration file, and the interface to external programs (for supplementing additional log facilities, like e.g. paging). There are also a few HOWTOs for specific issues (e.g. the client/server setup). This documentation is also available online.
- The distribution package includes man pages for the program and the configuration file, and sample configurations for Linux, FreeBSD, and Solaris.
- Furthermore, there are a couple of regression test scripts that are driven by a main script called `test/test.sh`. Run this without arguments to get some help.