Osiris 2.4.2-release
User Guide

What is Osiris?

Osiris is a host integrity management system that can be used to monitor changes to a network of hosts over time and report those changes back to the administrator(s). Currently, this includes monitoring any changes to the filesystems. Osiris takes periodic snapshots of the filesystem and stores them in a database. These databases, as well as the configurations and logs, are all stored on a central management host. When changes are detected, Osiris will log these events to the system log and optionally send email to an administrator. In addition to files, Osiris has the ability to monitor of other system information including user lists, group lists, and kernel modules or extensions.

Some integrity management systems are signature-based, that is, they look for specific file attributes as a means of detecting malicious activity. This type of approach to host integrity can be very cumbersome to manage and can lead to unwanted change going undetected.

Osiris is intentionally not like this. Osiris will detect and report changes to a filesystem and let the administrator determine what (if any) action needs to take place. There are no complicated assumptions or dumbing-down of information. If a change occurs, it can be detected and reported.

All Osiris components compile and run on both Windows and common UNIX systems, including BSD, Linux, Mac OS X and Darwin, and Windows NT/2K/XP. This allows for the flexibility to manage all types of platforms from either a UNIX or Windows environment.

Overview of the Components

Osiris consists of three major components: the management daemon (osirsimd), a scanning daemon (osirisd), and a management application (osiris). At the present time, the management application uses only a command-line interface.

osiris ↔ osirismd ↔ osirisd

The management daemon should be deployed on a trusted host. This is where all of the information about managed hosts are kept, including configs, logs, and databases. This process runs as an unprivileged user.

The scanning daemon is a lightweight process that runs on each monitored host. The scanning daemon is responsible for scanning the local filesystem and sending the data back to the management host. On UNIX systems, as well as Linux, the scanning daemon runs with privilege separation. The root process exists only to open privileged files during a scan. (files are never opened for write

access, never).

The management application is used by the administrator to manage the details of the scanned hosts. It communicates directly only with the management daemon.

The management daemon has a very simple, portable, directory structure and can be tar'ed or zipped and moved to any other supported system (if needed). It is organized as follows:

```
osiris/
  |
  |
  `--certs/               - root cert (osirismd.crt)
  |
  `--configs/             - default and shared scan config files
  |
  `--filter.db            - comparison filter database
  |
  `--hosts/               - host directories (see below)
  |
  `--osirismd.conf        - management daemon config
  |
  `--private              - user database, SSL key file
```

Each host directory under the "hosts" directory has the following structure:

```
example-host/
  |
  |
  `--configs/             - scan config files
  |
  `--databases/           - scan database files
  |
  `--logs/                - comparison log files
  |
  `--host.conf            - config for this host
```

Logging into the Management Daemon

The first thing to do after installation is configure the management daemon. The default configuration only allows connections from localhost, so you must do the initial configuration from the local machine.

Run the management application with no arguments (on Windows, run "osiris.exe" from a command prompt). By default, it will connect to the local management daemon. The first time you connect, you will be prompted for verification of the management daemon's certificate, e.g.:

```
bash-2.05b$ osiris
unable to load root cert for management host:
(/usr/home/brian/.osiris/osiris_root.pem)
fetching root certificate from management host (localhost).

The authenticity of host 'localhost' can't be established.

  [ server certificate ]

      subject = /C=US/CN=Osiris Managment Daemon/OU=Osiris IDS
      issuer  = /C=US/CN=Osiris Managment Daemon/OU=Osiris IDS

           key size: 2048 bit
    MD5 fingerprint: 46:8A:2D:63:92:C8:E9:E6:A8:36:5A:4D:23:E1:31:7F

Verify the fingerprint specified above.
Are you sure you want to continue connecting (yes/no)?
```

If accepted, the management application will store the presented root certificate in your ~/.osiris/ directory.

The management application currently only uses a command-line interface. To see a listing of commands, use the *help* command. To see help for any command use: help <command>

Many commands accept a host argument. Using the host command allows all subsequent commands to apply to the specified host. The prompt changes to show the currently chosen host in [] brackets. For example, the following prompt has the host named "local" selected:

```
osiris-2.0.0-[local]:
```

Configuring the Admin User

With this release, there is no concept of a user hierarchy. That is, all users are considered equal and are not limited in any way. Initially, there is only a single user: the admin user. By default this password is blank. One of the first things that should be done is to set the password for this user with the *passwd* command. For example:

```
osiris-2.0.0: passwd admin
Password:
user: (admin) updated.
current login was edited, you must re-authenticate.
authenticating to (localhost)

Password:

connected to management daemon, code version (2.0.0).
initial release

osiris-2.0.0:
```

To see a listing of all users, use the *users* command, e.g.:

```
osiris-2.0.0: users

  [ name ]

  admin

total: 1 users.
osiris-2.0.0:
```

Initial Setup and Configuration

Before hosts can be managed, it is a good idea to configure some settings for the management daemon. The defaults are practical, but some settings need to be customized . Use the *edit-mhost* command to kick off the configuration process. Here is an example:

```
osiris-2.0.0: edit-mhost

[ edit management host (localhost) ]

  > syslog facility [DAEMON]:
  > syslog level [INFO]:
  > log intensity [LOW]: medium
  > control port [2266]:
  > http port [2267]:
  > notification email []: bob@example.com
  > notification smtp host [127.0.0.1]:
  > notification smtp port [25]:

  > authorized hosts:

   127.0.0.1

  Modify authorization list (y/n)? [n] y

    a) Add a new authorized host.
    r) Remove authorized host.
    l) Show current listing.
    e) Exit

    > a
    > authorized hostname/IP (*=wildcard): 10.10.0.1

    a) Add a new authorized host.
    r) Remove authorized host.
    l) Show current listing.
    e) Exit

    > e

[ management config (localhost) ]

syslog_facility = DAEMON
syslog_level = INFO
log_intensity = MEDIUM
control_port = 2266
http_port = 2267
notify_email = bob@example.com
notify_smtp_host = 127.0.0.1
notify_smtp_port = 25
allow = 127.0.0.1
allow = 10.10.0.1
```

```
Is this correct (y/n)? y
management host config sucessfully saved.
osiris-2.0.0:
```

The above example accepted all of the defaults, and changed the access control list so that the management application can connect from host 10.10.0.1 as well the default localhost.

syslog_facility and *syslog_level:* used only if the management daemon is running on a UNIX host. For NT/2K/XP, this is ignored. The management daemon always logs to syslog, including database comparison results. This cannot be turned off. On Windows, these logs are sent to the “Applications” section of the event viewer.

log_intensity: setting can be one of: low, medium, high. The low setting will only log critical events. The high setting will turn on verbose logging.

control_port: the port that the management daemon will listen on. At the present time, this is not easily configurable because the management application is set to connect to that port. The setting is in there so that eventually it can be easily configurable in case the default port doesn't play with your network or firewall settings.

http_port: used to allow web based acknowledgment of changes to a host. When this port is set, any notification messages sent out will contain a special URL that can be used to confirm and accept the specified changes. This is only for convenience. To disable this feature, set the port value to zero.

The notify settings are for the notification subsystem. Notifications are currently only in the form of email. The *notify_email* setting is the address to mail alerts and messages to. The *notify_smtp* host is the mail server to use, and the *smtp_port* is the port to connect to. A quick way to disable notifications is to blank out the *notify_email* field. To test the notification address, use the ‘test-notify’ command.

Finally, the *allow* fields are the hosts that the management daemon will accept requests from. If a connection is made to the management port from an address not in this list, the connection is immediately dropped. To allow connections from any host, add an allow entry of "*".

Adding a Host

To manage a host, use the *new-host* command. This will prompt for some basic information, and optionally setup a trusted database for the new host. Here is an example of configuring localhost to be scanned:

```
osiris-2.0.0: new-host

[ new host ]

  > name this host []:  local
  > hostname/IP address []: 127.0.0.1
  > description []: local osiris scanner
  > host type (generic/router) [generic]:
  > enable scan logging for this host? (yes/no) [no]: yes
  > archive scan databases for this host?  (yes/no) [no]:
  > enable admin email notification for this host? (yes/no) [no]: yes
  > send scan notification, even when no changes detected (yes/no) [no]:
  > configure scan scheduling information? (yes/no) [no]: yes

    [ scheduling information for local ]

    Scheduling information consists of a start time and a frequency value.
    The frequency is a specified number of minutes between each scan,
    Starting from the start time.  The default is the current time.
    Specify the start time in the following format: mm/dd/yyyy HH:MM

    enter the start date and time: [Wed Jul 16 22:29:47 2003]
    enter scan frequency in minutes: [daily (1440)]

  > activate this host? (yes/no) [no]: yes

host                  => local
hostname/IP address   => 127.0.0.1
description           => local osiris scanner
host type             => generic
log enabled           => yes
archive scans         => no
notifications enabled => yes
notify_email          => (management config)
notifications always  => no
scans starting on     => Wed Jul 16 22:29:47 2003
scan frequency        => daily (every 1440 minutes).
enabled               => yes

Is this correct (y/n)? y
the new host (local) was sucessfully created.
```

The *name* field should be a short name because it will be used to reference the host in other commands.

The *hostname* field can be an IP address or hostname of the host to be scanned. This host must have a scanning daemon installed.

the *description* is just a short description of the host used only for display purposes.

host_type field should be set to "generic" for regular UNIX and windows hosts and "router" if the host is a router. Currently, the only supported host type is generic.

the *log_enabled* field, when set, creates a log file for each database comparison that is executed. Even if no changes have occurred, a log file is generated.

When the *archive_scans* field is set, new databases are created with each scan. For some hosts, it may be important to keep every scan database. Keep in mind, however, that depending upon how often the host is scanned and what is scanned, the database directory for that host can grow rather large. If this field is set to "no", then only the trusted database and the database from the last scan are kept.

The *notifications_enabled* field, when set, will notify the administrator with scan logs and schedule errors.

If the *notify_email* field is set, notifications for this host is sent the the email address specified. Otherwise (if it is left blank) the email address set in the management configuration is used. If notifications are not enabled, this setting has no effect.

The *notifications_always* field, when set, will notify the administrator even in the case of empty scan logs. This can be useful if you require confirmation that the scanned host has not changed (according to its config). If set to "no" then the administrator is not notified when there are no changes; only when there are changes. If notifications are not enabled, this setting is ignored.

Scheduling for scans is fairly simple. There is a start point and a periodic interval (in minutes) for when scans are conducted. The scheduled scan start time is used as a reference point for the initial scan. In the above example, this host will be scanned every a day at 11:29 p.m.

Finally, the *enabled* field is for the quick enabling/disabling of a host. If a host is disabled, the scheduler will ignore it. The admin can still kick off scans manually, but no automated scans will occur.

Initializing a Host

As the last step of adding a host, you will be asked to initialize the host. What this means is that the management daemon will import a default config for the platform, push it to the scanning daemon, start a scan to create an initial database, and set this database to be the trusted database:

```
initialize this host? (yes/no): y

Initializing a host will push over a config, start
a scan, and set the created database to be the trusted
database.

Are you sure you want to initialize this host (yes/no):

OS Name: Darwin
OS Version: 7.0.0b1

use the default config for this OS? (yes/no): y
default config for (Darwin) has been pushed.
perform an initial scan and database for this host? (yes/no): y
scanning process was started on host: local
```

Upon examination, you can verify that the database was created with the "databases" command:

```
osiris-2.0.0[local]: databases local
This may take a while...

  [ name ]                        [ created ]

  * 1                               Wed Jul 16 23:01:48

total: 1
(*) denotes the base database for this host.

osiris-2.0.0[local]:
```

You can put a host through the initialization process at any time by using the *init* command.

Scan Databases

When a host is scanned, all scan data is sent back to the management host. None of the information collected is ever kept on the filesystem of the managed host. This is done in an attempt to preserve the integrity and confidentiality of the scan data.

The management host stores scan data in Berkeley DB files. Each scan database contains some meta data, the scan entries, and any errors that may have occurred during the scan. The databases are formatted in such a way that they can be moved across platforms or machines with different byte ordering. This makes it easy to migrate a management host to a different platform with little hassle. This is also handy in a forensic situation by allowing a timeline of scans to be analyzed on any of the supported platforms.

The scan database files are considered read-only and are even created so as to not allow write privileges to any user. Thus, database files are NEVER altered in any way by any of the Osiris executables.

Each host has one database that is considered the base, or trusted, database. When a subsequent database is created, it is compared against the base database in order to determine if any changes have occurred. Since databases are never altered, change management is handled by changing which database is the trusted database. More information on this in the section below, "Dealing with Detected Changes".

Hosts can be configured to have database archiving enabled or disabled. In most cases, it is sufficient to disable archiving, but your security policy might require that each database be kept and stored for some period of time. If archiving is enabled, keep in mind the scan frequency and size of each database when determining disk space requirements.

The following table lists the file attributes scanned with each scan record type:

SCAN_RECORD_WINNT - Windows NT/ 2K / XP

```
path
cryptographic checksum
owner name
owner SID
group name
group SID
modification time
access time
change time
file size (bytes)
device
attributes (archive,compressed,dir,encrypted,hidden,normal,indexed,
offline,readonly,reparse,sparse,system,temporary)
```

SCAN_RECORD_UNIX1 - FreeBSD, Mac OS X, BSD/OS, OpenBSD, Linux

```
path
cryptographic checksum
permissions string
user name
group name
device
inode
file mode
number of hard links
uid
gid
modification time
access time
change time
device type
file size (bytes)
blocks
block size
```

Viewing Database Contents

At times, it may be necessary to view the contents of a scan database. The command line interface allows for this with the 'print-db' command. Use the 'list-db' command to view a listing of databases. The 'print-db' command will then enter a subcommand mode for that database that allows for the printing of the database header, file record listing, specific file details, system record listing, and any errors that were encountered while scanning. For example:

```
osiris-3.0.0-dev[local]: list-db
This may take a while...
```

```
  [ name ]                          [ created ]

  * 1                                 Mon Jan  5 07:28:55
    2                                 Tue Jan  6 07:21:17

total: 2
(*) denotes the base database for this host.

osiris-3.0.0-dev[local]: print-db 2
This may take a while...

100% [=========================================>] 233472 bytes

    h) show database header.
    r) list file records.
    d) list file record details.
    s) list system records.
    x) list errors.
    q) quit

[local:database: 2]: h

  DATABASE: 2

         status: complete
           host: unknown
           user: unknown
         config: test (35c8e80b)

         errors: 0
   file records: 34
 system records: 55

  SCAN RESULTS:

            record type: UNIX1

      files encountered: 34
          files scanned: 34

   symlinks encountered: 0
      symlinks followed: 0

       files unreadable: 0
 directories unreadable: 0
    symlinks unreadable: 0

           scan started: Tue Jan  6 07:21:17 2004
          scan finished: Tue Jan  6 07:21:18 2004


    h) show database header.
    r) list file records.
    d) list file record details.
    s) list system records.
    x) list errors.
    q) quit
```

```
[local:database: 2]:
```

How Scheduling Works

The built-in scheduler is responsible for telling the managed hosts when to start a scan. When it is time for a host to be scanned, the scheduler does the following:

1. Sends the scan config used to create the currently trusted database to the managed host.

2. Starts a scan on the remote host.

3. When the scan is complete, the newly created database is then compared to the trusted database. Depending upon the host's configuration and the results of the database comparison, the comparison results might or might not be sent to the administrator.

The scheduler pushes the scan config each time to ensure that the managed host has the correct scan config loaded. The current scheduler is limited to scheduling scans based off of a single config.

Dealing With Detected Changes

The use of Osiris can vary in purpose, but the bottom line is that the administrator desires to be notified of certain changes and not others. Whatever the case, there are always unanticipated changes. Software is upgraded. Security alerts trigger patches to critical parts of a system.

When the latest scan produces information that differs from the trusted state, log entries are created and email might be sent to the administrator listing the detected changes. Unless action is taken, the subsequent scans will report the same changes and, depending upon the host's configuration, continue to notify the administrator.

Since databases are not altered, the way to "acknowledge" changes is to change what is considered the trusted database for that host. There are two ways to do this:

1. Log in to the management host and use the *set-base-db* command to set the base database. Usually this will be the most recently created database. When managing a lot of hosts, this can be cumbersome.

2. Enable email notifications and the http server port. When a change occurs for a host, email is sent to the administrator listing the changes.

Included in each email is a URL that will allow the administrator to use a web browser to quickly command the management host to change the trusted database accordingly. The same authentication logic applies. The user must login from a host that is in the management host's allowed access list.

Notifications

Currently, the only supported notification is email. Although monitoring of the logs is strongly recommended, it can be very convenient to be notified of changes by email. Hosts can be configured to notify the administrator after each scan, or only when changes are detected. A warning notification is issued in the event that a scheduled scan fails for any reason.

A notification consists of statistics related to the scan as well as all of the log entries associated with any detected changes. If the http server is enabled, a URL is also specified to provide the recipient the opportunity to update the trusted database via an SSL enabled web browser.

The following is an example of a notification after updating some executables:

```
If these changes are approved, visit the URL below to set the
latest scan database to be the trusted database. Or, login to the
management console and set the trusted database to 4.If these
notifications persist, you may need to modify the scan config for
this host.

    <https://example.com:2267?host=mac-ibook&base_db=4>


Change Statistics:
-----------------------------------

        checksums: 1
       SUID files: 0
 root-owned files: 3
 file permissions: 0
        new files: 0
    missing files: 0

total differences: 10

     compare time: Fri Sep 19 23:27:37 2003
             host: mac-ibook
         log file: no log file generated, see system log.
          base db: 3
       compare db: 4

[mac-ibook][cmp][/usr/sbin/osiris][mtime][Fri Sep 19 14:32:00
2003,Fri Sep 19 23:22:51 2003]
[mac-ibook][cmp][/usr/sbin/osiris][ctime][Fri Sep 19 14:32:00
2003,Fri Sep 19 23:22:51 2003]
[mac-ibook][cmp][/System/Library/StartupItems/Osiris][mtime][Fri
Sep 19 14:32:01 2003,Fri Sep 19 23:22:51 2003]
[mac-ibook][cmp][/System/Library/StartupItems/Osiris][ctime][Fri
Sep 19 14:32:01 2003,Fri Sep 19 23:22:51 2003]
[mac-ibook][cmp][/usr/sbin/osirisd][mtime][Fri Sep 19 14:32:00
2003,Fri Sep 19 23:22:51 2003]
[mac-ibook][cmp][/usr/sbin/osirisd][ctime][Fri Sep 19 14:32:00
2003,Fri Sep 19 23:22:51 2003]
[mac-
ibook][cmp][/usr/sbin/osirismd][checksum][1df0207690badc70f5503a729
3514a77b877101,cfcd60e2cea349b8223703b3d16fd06b6523b17]
[mac-ibook][cmp][/usr/sbin/osirismd][mtime][Fri Sep 19 14:32:00
2003,Fri Sep 19 23:22:51 2003]
[mac-ibook][cmp][/usr/sbin/osirismd][ctime][Fri Sep 19 14:32:00
2003,Fri Sep 19 23:22:51 2003]
[mac-ibook][cmp][/usr/sbin/osirismd][bytes][1575356,1575460]
```

Comparison Filters

Sometimes it may be necessary to monitor only certain attributes of files. For

example, it is common for log files to change in size and content but the permissions should probably never change.

Comparison filters allow certain changes to exist without triggering a log entry. One of the biggest problems with IDS systems is that they often become noisy, or issue bothersome alerts that eventually numb the administrator to the point where events are ignored or not given proper attention. Comparison filters allow an administrator to specifically single out changes that should be ignored, or changes that should given attention so that notifications become more meaningful.

These filters can be configured per host, and can be applied to new or missing files as well as any of the file's attributes. Regular expressions can be used to filter out multiple files. Regular expressions can also be used to apply a filter to multiple hosts.

Currently, comparison filters cannot be applied to the system monitoring facilities (users,groups,kerenel extensions).

To list the current comparison filters, use the 'filters' command. Use the 'edit-filters' command to manage filters. The following example illustrates the creation of a comparison filter to exclude all changes to a log file except permissions and ownership:

```
osiris-3.0.0: edit-filters

    s) show current filters.
    a) add a new filter.
    r) remove filter.
    e) exit

    > a

  > host (*=all hosts): *
  > path (*=any path): /var/log/secure.log

  1) Exclude (ignore changes to certain attributes)
  2) Include Only (monitor changes only to certain attributes)

  > filter type: [2] 2
  csum
  device
  inode
  perm
  links
  uid
  gid
  mtime
  atime
  ctime
  dtype
  bytes
  blocks
  bsize
```

```
  osid
  gsid
  new
  missing

  > attributes (comma separated): perm,uid,gid,missing

  does this look correct:
    ==> host=*;path=/var/log/secure.log;include only: perm uid gid missing ;
(y/n)? y

filter added.

    s) show current filters.
    a) add a new filter.
    r) remove filter.
    e) exit

    > e
comparison filter list sucessfully saved.
osiris-3.0.0:
```

Scan Configuration Syntax

Scan configs are used to dictate exactly what is scanned on a host. Osiris ships with default configs for many platforms, so you have a decent starting point.

The config syntax for Osiris is very similar to the config sytnax used by Apache. A scan config consists of a global section, and a list of blocks. Each block represents a directory and contains options for that block, as well as rules used to include or exclude certain files.

Blocks are specified within directory tags:

```
<Directory></Directory>
```

If an option is not specified, the global value is assumed. If no rules are listed then the default rules are assumed. If a block is a subdirectory of another block, the subdirectory block takes precedence.

Options

Options dictate the nature of the scanning that takes place in the directory. Usually options appear first in the block, but their placement within the block has no signifigance or precedence. Options not found within a specific block are said to be global. If no global value is specified, there is a default value for each option.

The list of currently valid options are as follows:

- Recursive <bool>

  if set, this osiris will recursively scan this directory, otherwise, only a top level scan will be done.

- FollowLinks <bool>

  if set, osiris will follow any symbolic links it encounters. If a symbolic link is a link to a directory, it will only follow that link if the the recursive option is set.

- Hash <hash>

  specifies the hash algorithm to be used for all files in the directory. algorithm may be one of: md5, sha, or ripemd

  boolean values must be one of: yes, y, 1, true, yup, no, n, 0, false, nope

hash value must be one of: md5, sha, haval, ripemd

Rules

a rule serves to include or exclude files, either specifically, or by property. The order of the rules is important. When a file is encountered during a scan, the rules for that block are analyzed in the order they are listed. If a rule catches a file, the information about the file is stored in the resultant database. Otherwise,
it is ignored.

The list of rules and their formats are as follows:

- IncludeAll

  Include all files, all files will be caught under this rule. If a block contains no rules, this is the default rule assumed.

- ExcludeAll

  Exclude all files.

- Include <filter>

  Include any files that pass the specified filter.

- Exclude <filter>

  Exclude any files that pass the specified filter.

- NoEntry <directory>

  This only applies to recursive blocks. The specified directory is not entered and the directory file itself is also ignored. The directory specified should be relative to the block. For example, to exclude the contents of /tmp as well as the /tmp directory file, specify the following:

```
<Directory />
   NoEntry /tmp
   Exclude file (“/tmp$”)
</Directory>
```

Filters

In every block of the scan config, including the global block, there will at be at least one filter. If no filters are specified in a block, the global filter will be used.

Filters are found as part of a rule. Specifically, you include filters after an 'Include' or 'Exclude' rule. For example, the following two statements include filters that will catch, or include, the files that are perl scripts or any file owned by user bob:

```
Include perl
Include user("bob")
```

Filters are designed to catch files of a particular type, or files that share a specific attribute. The information used to create many of the filters found in osiris are derived from the unix file(1) utility. The following is a list of the currently implemented filters, and their official definitions:

- sticky

  any directory or file that has the sticky bit set. For operating systems that do not support the sticky bit, this filter does not apply and has no effect.

- suid

  any file that has the suid bit set.

- sgid

  any file that has the sgid bit set.

- executable

  All ELF binaries.

  FreeBSD/OpenBSD - all files that begin with any of the following 4 byte long expressions:

  (4 bytes) & 0377777777 = 041400407
  (4 bytes) & 0377777777 = 041400410
  (4 bytes) & 0377777777 = 041400413
  (4 bytes) & 0377777777 = 041400314

  BSDi - all files that begin with any of the following bytes:

  0xCC, 0x107, 0x108, 0x10b

Linux - all files that begin with any of the
following 4 byte long values:

0x00640107
0x00640108
0x0064010b
0x006400cc

or

\01\03\020\04
\01\03\040\04

Solaris/SunOS - all files that begin with any of the
following 4 byte long expressions:

(4 bytes) & 077777777 = 0600413
(4 bytes) & 077777777 = 0600410
(4 bytes) & 077777777 = 0600407
(4 bytes) & 077777777 = 0400413
(4 bytes) & 077777777 = 0400410
(4 bytes) & 077777777 = 0400407
(4 bytes) & 077777777 = 0200413
(4 bytes) & 077777777 = 0200410
(4 bytes) & 077777777 = 0200407

Darwin

(4 byte header: feadface)

- perl

any file that contains, within the first thirty bytes, any of the following
strings: "/bin/perl", "/usr/bin/perl", "/usr/local/bin/perl".

- python

any file that is a python script, a python text executable, or a python
compiled file. Specifically any file with any of the following attributes:

- first three bytes: \032\032\032
- first four bytes: \010\013\078\153 (little endian)

- script

  any shell script or script, including sh, csh, bash, ksh, tcsh, ksh, ash, ae, nawk, gawk, awk, rc, env. The file must begin with the string: "#!" or "BEGIN" and must contain one of the following within the first 25 bytes:

  ```
  /bin/sh
  /bin/csh
  /usr/local/bin/bash
  /bin/bash
  /bin/ksh
  /bin/tcsh
  /usr/local/tcsh
  /usr/local/bin/tcsh
  /usr/local/bin/zsh
  /usr/local/bin/ash
  /usr/local/bin/ae
  /bin/nawk
  /usr/bin/nawk
  /usr/local/bin/nawk
  /bin/gawk
  /usr/bin/gawk
  /usr/local/bin/gawk
  /bin/awk
  /usr/bin/awk
  /bin/rc
  /usr/bin/env
  ```

- gzip

  any GNU zipped file or jar file, specifically, any file that begins with the following two bytes:

  ```
  \037\213
  ```

- zip

  any file created with winzip, or zip, specifically, any file that begins with the first four bytes:

  ```
  PK\003\004
  ```

-

tar

any file created with tar, or GNU tar utilities, specifically, any file that contains the string "ustar" at byte positions 257-261.

- pgp

any PGP public keyring, security ring, encrypted data, ascii armored data, public key block, message, signed message, or signature. Specifically any file that begins with any of the following two bytes (little endian):

0\153
1\153
1\149
0\149
0\166

or the follwing string:

-----BEGIN\040PGP

- rpm

any rpm package file, specifically, any file that begins with the following four bytes:

\237\171\238\219

- uid(x)

any file that is owned by the uid: x

- gid(x)

any file that is a member of the group: x

- sid(x)

Windows only: any file that is owned by the specified SID

- user(x)

any file that is owned by user: x

- group(x)

  any file that is a member of the group: x

- header(x)

  any file that begins with the literal hex value: x

- file(regex)

  any file that has the name matching the regular expression specified. Paths can be relative to the block's directory. For example, in the block "/etc", the filter:

  file("motd")

  actually refers to "/etc/motd". Likewise, specifying:

  file("rc.d/init.d")

  will match the file "/etc/rc.d/init.d". For platforms that do not have native support for regular expressions, it is recommended that the full path to the file be specified.

- suffix(x)

  any file that has a name with the format: *.x

- md5(x)

  any file that has an MD5 checksum value of: x

- sha(x)

  any file that has an SHA-1 checksum value of: x

- ripemd(x)

  any file that has an RIPEMD-160 checksum value of: x

- permissions(x)

  any file that has a permissions string with the format: x. The character '*' can be used for a wildcard to denote that any value for that bit is acceptable. Substrings can also be used. For example, to specify all files that have owner read bit set, use: "*r" as a value. To speicfy all files with suid bit set,

use: "***s", this is essentially the same as using the suid filter.

The Global Block

The global block does not apply to any specific directory and is not within any tags. All options are valid in the global block. The only valid rules in the global block are IncludeAll and ExcludeAll rules.

The Default Configuration

Global options are assumed if not specified. However, if no global option is specified, the default values for options are as follows:

```
Recursive       yes
FollowLinks     no
Hash            md5
```

Example Scan Config

The following is the included default config for Mac OS X. It scans most of the executable
directories as well as root's home and the NetInfo database files.

```
Recursive   no
FollowLinks no

IncludeAll
Hash sha

<Directory /private/var/root>
    Recursive yes
    Include executable
    Include suid
</Directory>

<Directory /bin>
</Directory>

<Directory /usr/bin>
</Directory>

<Directory /usr/local/bin>
</Directory>

<Directory /sbin>
</Directory>

<Directory /usr/sbin>
</Directory>

<Directory /var/db/netinfo>
</Directory>
```

Custom Scan Configs

Osiris ships with default scan configs, but you will likely want to modify those or create your own.

The "configs" directory in the osiris root on the management host contains all of the default configs, e.g. “default.linux”. These are the configs used during the init process and when a host is created. Feel free to modify these files so that the addition of new hosts can take advantage of the custom configs. The config files in this directory are known as “shared” because any number of hosts can take advantage of them. This makes it easy to monitor many similar hosts and be able to make changes to their scan by editing a single config file.

At any point in time, you can view the config used to create the trusted database for a host, use the ‘config’ command:

```
osiris-2.2.0: config mac-ibook
the trusted database for host: mac-ibook created with config: (default.darwin)
```

For hosts that require very customized configs, you may choose to store it under that host’s configs directory. The only real advantage of this is that the main ‘configs’ directory will not get cluttered with customized configs.

To create a new config, use the ‘new-config’ command. To create a config in the shared configs directory, just provide a name:

```
new-config myconfig
```

Or, to add a config to be used only for a specific host:

```
new-config myhost myconfig
```

Once a config has been created, it can be edited with the 'edit-config' command. This command will download the config and open it in an editor on the local machine. If saved, the modified config will be sent back (saved) to the management host. Note that editing the config will most likely generate a different database signature and generate log entries.

To list all of the configs, use the ‘configs’ command. To verify that a config is syntactically valid, use the ‘verify-config’ command. The verify command will also print any errors and their line numbers.

Logging

System log

The management daemon logs to the system log (syslog on UNIX platforms and the Event Viewer on Windows). Syslog parameters (facility and level) are specified in the management daemon's configuration file.

There are three levels of logging: low, medium, high. The low setting has only critical events sent to the log. The high setting is the most verbose.

Log Files

If a the *log_to_file* attribute is enabled for a host, a log file is created after each scan. The log details the results of the comparison. Comparison logs are always sent to the system log regardless of the log level. This cannot be turned off.

Scanning daemons do not log anything related to scans to the system log, only events related to the status or state of the daemon itself. This is by design. The goal is to keep sensitive information in a central location for reasons of security and usability.

Log Format

In an attempt to make these log events easier to parse or analyze, the following format is used for all log entries, including comparison logs:

```
[host][type] message
```

If there is no relevant host for the message, the host field will contain an "*". The valid entry types are:

```
info - informational message
err - error occured
warning - something that might warrant attention
cmp - Comparison related (something has changed)
```

For example, here is an informational log entry:

```
Jul 20 21:05:41 localhost osirismd[1461]: [*][info] authenticating
127.0.0.1...
```

The message field is usually a descriptive sentence. Comparison log entries have the following message format:

```
[path][attribute][trusted,current]
```

The path is a full path of the file in question. The attribute is the attribute that has changed (e.g. inode, mtime, or checksum). The third field contains two values: the trusted or initial value for the attribute and the current value separated by a comma.

For example, here is a comparison log entry that shows that the mtime value for "/bin/ln" has changed:

```
      [local][cmp][/bin/ln][mtime][Sat Jun 14 09:32:29 2003,Fri Jul 18 01:08:44
2003]
```

Recommendations

Osiris is meant to be a flexible tool. It can be used to monitor a lab full of similar hosts, or a few critical hosts, or maybe just a single box. Whatever your situation, it is important to know how to use this tool effectively. Therefore, we have assembled this short section of recommendations; feel free to follow or ignore them.

- Don't scan the entire drive! It will take forever, create massive databases, and hog system resources just like those stupid virus scanners. This tool wasn't designed to monitor every file on your box(es).

- Use the default configs, or stick to scanning executables and other critical files, such as libraries, system files, and kernel files. It is most common to use this tool to be made aware of changes to critical parts of systems. This keeps notifications down to a minimum.

- Use MD5 instead of SHA-1 as a checksum algorithm. It will allow for much quicker scans.

- Use a separate og analysis tool. Interpreting the entire change-set is not one of this tool's stronger features. Osiris logs will list exactly what has changed since the last scan, and not make guesses as to what happened; this is a job for a different tool, or an admin staring at the logs. By using another log analysis tool (such as swatch), the notification sub-system can be turned off –less services running, less to go wrong. All osiris logs are designed so they can be easily parsed.

- Don't create a million comparison filters. If you are getting bombarded with notifications, it's possible you need to adjust what files are being scanned. Comparison filters should be used carefully. It is possible you may forget about certain filters and unintentionally filter out important changes.

- Start with reasonable schedules. Don't initialize a box and tell it to scan every 5 minutes with an untested scan config. You may get a lot of email notifications. The default configs were designed to monitor files that (in most cases) will not regularly change. After running with a config for a while, then increase the scan period.

- Don't archive databases or logs (the default). If you need them, by all means keep and archive them for safe-keeping. Otherwise, they just take up disk space.

Further Information and Support

Osiris is still under active development. We encourage your comments, suggestions, or feedback on this product. Our goal is to produce an open and trusted host integrity management system that is free from the cumbersome maintenance requirements typically found with these types of tools.

Documentation and downloads are available here:

http://osiris.shmoo.com

To get support for using Osiris, or to interact with Osiris developers, use the following lists:

osiris@shmoo.com
osiris-devel@shmoo.com

To signup for these lists, visit:

http://osiris.shmoo.com/lists.html

Or, feel free to send mail to the lead developer: brian@shmoo.com