

# SNARE

System iNtrusion Analysis & Reporting Environment

## Guide to Snare for Linux

INTERSECT  
ALLIANCE

## Documentation History

Version No.	Date	Edits	By whom
1.0	16 November 2003	First draft for the Guide to Snare for Linux documentation.	Leigh Purdie
1.1	23 August 2004	Minor updates associated with sample script, and auditable system calls.	Leigh Purdie
2.0	2 April 2005	Minor rewording.	Leigh Purdie
2.1	30 November 2005	Formatting changes	George Cora
2.2	23 March 2006	Minor wording changes	Leigh Purdie
3.0	8 December 2006	Major update for Snare for Linux v1.0+	Leigh Purdie
3.1	24 April 2007	Updates for new features	David Mohr
3.2	18 June 2008	Updates for new features	David Mohr

© 1999-2008 Intersect Alliance Pty Ltd. All rights reserved worldwide.

Intersect Alliance Pty Ltd shall not be liable for errors contained herein or for direct, or indirect damages in connection with the use of this material. No part of this work may be reproduced or transmitted in any form or by any means except as expressly permitted by Intersect Alliance Pty Ltd. This does not include those documents and software developed under the terms of the open source General Public Licence, which covers the Snare agents and some other software.

The Intersect Alliance logo and Snare logo are registered trademarks of Intersect Alliance Pty Ltd. Other trademarks and trade names are marks' and names of their owners as may or may not be indicated. All trademarks are the property of their respective owners and are used here in an editorial context without intent of infringement. Specifications and content are subject to change without notice.

## About this guide

This guide introduces you to the functionality of the Snare Agent for the Linux operating system. Snare for Linux provides a C2-style auditing subsystem for the Linux operating system, and facilitates objective-based filtering, and remote audit event delivery. Snare for Linux will also allow a security administrator to fully remote control the application through a standard web browser if so desired. Snare has been designed in such a way as to allow the remote control functions to be easily effected manually, or by an automated process.

Other guides that may be useful to read include:

- Snare Server User's Guide.
- Snare Server Installation Guide.
- Snare Server Troubleshooting Guide.
- The Snare Toolset - A White Paper.

### Table of contents:

<b>1 Introduction.....</b>	<b>4</b>
<b>2 Overview of Snare for Linux.....</b>	<b>5</b>
<b>3 Installing and running Snare.....</b>	<b>6</b>
3.1 Snare installation.....	6
3.2 Manual installation.....	7
<b>4 Setting the audit configuration.....</b>	<b>8</b>
4.1 Audit configuration.....	8
4.2 General Configuration.....	9
4.3 Remote Control Configuration.....	10
4.4 Objective configuration.....	11
4.5 Event Display.....	15
<b>5 Snare Server.....</b>	<b>16</b>
<b>6 About InterSect Alliance.....</b>	<b>17</b>
<b>Appendix A - Configuration File Description.....</b>	<b>18</b>
<b>Appendix B - Event Output Format.....</b>	<b>21</b>

## 1 INTRODUCTION



The team at InterSect Alliance have experience with auditing and intrusion detection on a wide range of platforms - Solaris, Windows NT, Windows 2000, Novell Netware, AIX, even MVS (ACF2/RACF); and within a wide range of IT security in businesses such as - National Security and Defence Agencies, Financial Service firms, Government Departments and Service Providers.

This background gives us a unique insight into how to effectively deploy host and network intrusion detection systems that support and enhance an organization's business goals.

The development of 'Snare for Linux' will allow event logs from the new native Linux audit subsystem to be collected from the operating system, and forwarded to a remote audit event collection facility after appropriate filtering. Snare for Linux will also allow a security administrator to fully remote control the application through a standard web browser if so desired. Snare has been designed in such a way as to allow the remote control functions to be easily effected manually, or by an automated process.

In the spirit of the release of the Snare agents, InterSect Alliance are proud to release Snare for Linux as an open source initiative. Other event audit modules for AIX, IRIX, Solaris, Windows and other applications have been released under the terms of the GNU Public License. The overall project is called 'Snare' - **System iNtrusion Analysis & Reporting Environment**. The '*Snare Server*' is a commercial release of software beneficial to organizations that wish to collect from a wide variety of Snare agents and appliances such as firewalls or routers.

InterSect Alliance welcomes and values your support, comments, and contributions. Our contact details are available from our contact page at [www.intersectalliance.com](http://www.intersectalliance.com).

## 2 OVERVIEW OF SNARE FOR LINUX

Snare operates through the actions of three complementary components:

- The native Linux audit subsystem, which may be enabled in kernels newer than 2.6.12, on many distributions of Linux.
- The user-space audit daemon (auditd) version 1.0.15 and newer, which is included in several modern distributions, including Red Hat Enterprise 4, CentOS 4, and Fedora Core 5.
- The Snare 'dispatcher' applications.

The audit daemon, and kernel component act in concert to extract events of interest from the operating system.

Snare for Linux operates as a 'audit dispatcher' application that receives the audit log data, selectively filters out events that you are not interested in, formats the resulting data into something that is more suited to follow-on processing, and delivers it to one or more remote systems over the network.

Snare formats the audit log data into a series of 'tokens'. Two different field separators are used in order to facilitate follow-on processing - TABS separate 'tokens', and COMMAS separate data within each token. This format is further discussed in the section on the Snare output format. The result is that a raw event, as processed by Snare, may appear as follows:

```
fc5.intersectalliance.com LinuxKAudit criticality,1 event,execve,20061117 23:16:27
uid,0,root gid,0,root euid,0,root egid,0,root process,2520,ls
return,0,yes a0,8d39f88 a1,8d27cd0 a2,8d28ff0 a3,8d40b78
arch,40000003 auid,0,root cwd,/var/log/audit dev,fd:00
dev:1,fd:00 exe,/bin/ls flags,101 flags:1,101 fsgid,0,root
fsuid,0,root inode,881304 inode:1,1403596 item,0 item:1,1
items,2 mode,0100755 mode:1,0100755 name,/bin/ls ogid,0,root
ogid:1,0,root ouid,0,root ouid:1,0,root rdev,00:00 rdev:1,00:00
sgid,0,root suid,0,root utime,1163765787.425
```

Snare also incorporates a tiny embedded web server, the Remote Control Interface, which allows administrators to remotely control which events are collected and reported. The Remote Control Interface also provides information on users, groups, and group membership on the local machine, which can be used to satisfy various regulatory security requirements.

Snare for Linux is known to work on Red Hat Enterprise 4 (Update 3), CentOS 4 (Update 3), and Fedora Core 5. With the modification of an audit startup script, SuSE 10.1 is also known to work. The team from Canonical are investigating potential options for Ubuntu.

## 3 INSTALLING AND RUNNING SNARE

### 3.1 SNARE INSTALLATION

Snare is available as an RPM package that enables it to be installed and removed with relative ease on systems that utilize RPMs. Snare is also available as a normal 'source archive' (gzipped TAR file). Third parties may also make other package formats available, such as Debian/Ubuntu DPKG.

- ▶ WHAT YOU NEED...**
- An appropriate Linux distribution.  
A distribution that has the audit capability turned on in its supplied kernel. At present, Red Hat Enterprise 4, CentOS 4, and SuSE 10.1 are known to support this capability.
  - audit version 1.0.15 or newer  
This provides the necessary binaries to funnel audit event information from the Linux kernel, into Snare via the "dispatcher" configuration option. This package will generally be provided by your Linux distribution vendor.
  - The SnareLinux package in RPM format, or a format appropriate for installation on your system.  
Snare for Linux provides the infrastructure required to filter, format and distribute audit log data to one or more central log collection systems. NOTE that the InterSect Alliance site provides binaries for several common distributions, but if your distribution is not supported, you may need to recompile & install *Snare* from either the source RPMs, or the basic 'tar archive'.

**▶ HOW TO...** Compile from a source RPM

To recompile, and install for your own system, try the following commands as root:

1. `rpm --rebuild SnareLinux-1.4-1.src.rpm`  
The software should compile. Near the end of the build text, a line similar to the following will appear:  

```
Wrote: /usr/src/redhat/RPMS/i386/SnareLinux-1.4-1.i386.rpm
```
2. Use this filename to install the new *SnareLinux* package:  
`rpm -Uvh /usr/src/redhat/RPMS/i386/SnareLinux-1.4-1.i386.rpm`

**▶ HOW TO...** Install Snare and SnareCore binary RPM packages.

Installation of the Snare package is reasonably straightforward:

1. Download the required RPMs, as above
2. Logon as root user, i.e. enter the command `/bin/su -` at the command prompt, and enter the root password when prompted. Issue the command, as root:  
`rpm -Uvh SnareLinux-1.4-1.i386.rpm`
3. Note that the audit daemon will restart after Snare has been installed.

---

## 3.2 MANUAL INSTALLATION

If you do not use the Red Hat Package Manager to facilitate installation, then retrieve the *SnareLinux* tar file, uncompress and untar it, and issue the following commands:

 **HOW TO...** Install from TAR files:

For *SnareLinux*:

1. make clean
2. make
3. make install

## 4 SETTING THE AUDIT CONFIGURATION

### 4.1 AUDIT CONFIGURATION

Once the Snare agent is installed, it will begin to operate using a very simple configuration.

The audit configuration is stored as `/etc/snare.conf`. This file contains all the details required by the audit daemon to successfully execute. Failure to specify a correct configuration file will not 'crash' Snare, but may result in selected events not being able to be read.

**Tip:** Manual editing of the `snare.conf` configuration file is possible, but care should be taken to ensure that it conforms to the required format for the audit daemon. Also, any use of the Remote Control Interface to modify security objectives or selected events, may result in manual configuration file changes being overwritten. Details on the configuration file format can be viewed in [Appendix A - Configuration File Description](#).

This configuration can either be changed by editing the configuration file, `/etc/snare.conf`, directly, or by modifying the objectives via the tiny web server embedded within Snare. This Remote Control Interface operates completely in memory, and does not rely on any external files, other than the `snare.conf` file. The Remote Control Interface component is not turned on by default.

#### ▶ How to... Remote Audit Monitoring

The Remote Control Interface can be turned on by tweaking the default `snare.conf` file. You can either edit the `/etc/snare.conf` file directly, uncommenting the 'allow=1' line under the '[Remote]' section, or you can execute the following command as root:

```
sed -i 's/^#[ \t]*allow=1/\tallow=1/' /etc/snare.conf
```

The Remote Control Interface provides a number of capabilities, broken up into the following general categories:

- General Configuration
- Remote Control
- Objectives
- Recent Events
- User and Group meta-data

## 4.2 GENERAL CONFIGURATION

The audit configuration parameters to consider are:

- The destination address(es) for audit events.

Snare for Linux can send audit events to one or more network destinations. Enter a DNS name, or IP address for each planned destination.
- Whether the events are going to a Snare Server, or a SYSLOG server.

Snare can send data either to a Snare-compatible server (which includes our free/GPL 'BackLog' application, as well as the full Snare Server appliance), or a SYSLOG-compatible destination. Please be aware that most SYSLOG servers operate on a 'single threaded' design, which generally means that they are incompatible with the extremely high volumes of data Snare is capable of generating if configured to monitor high-volume system calls (such as file opens).
- Whether you wish Snare to take control of your audit configuration.

By default, Snare will manage your audit event settings for you. Normally on a Linux system, you will need to modify the file `/etc/audit.rules` in order to establish a new monitored event. Snare has the capability to 'turn on' event auditing in response to the objectives you set within the Remote Control Interface. We recommend leaving this feature on.
- Whether the 'criticality' rating assigned by Snare to each event, is important to you.

By default, Snare assumes that the 'criticality' rating you give to each objective, is important. As such, rather than stopping at the first objective 'match' for an event, Snare will continue through each objective, and report the criticality value for the 'most important' match. If you do not require the criticality value, turning off this feature will result in a noticeable reduction in Snare's resource usage.
- What SYSLOG destination you wish to send Snare events to (if applicable).

If you are sending your data to a SYSLOG server, you may wish to alter the SYSLOG destination values to something appropriate for your system.
- What enterprise features to enable. Note that the following options are only available to users who purchase a Snare Server. These are not part of the Open Source toolset. See Chapter 8 below for more details on the supported versions of the Snare agents.
  - Use UDP or TCP - Select the protocol you would like Snare to use when sending events. Using TCP will guarantee message delivery.
  - Cache size - Allow Snare to store messages that could not be sent. Combined with the TCP, this option will allow the agent to cache messages if there is a network failure or the Snare Server is otherwise unavailable. Any cached message is kept (even if the agent is restarted) until it is sent or the size of the cache exceeds the specified allotment, in which case the oldest message is removed.
  - Encrypt Message - Encrypt messages between the agent and the Snare Server. This option requires matching Remote Access Passwords on both the agent and the Snare Server.

These parameters are shown in the 'General Configuration' link in the Remote Control Interface of the Snare for Linux agent, shown below.

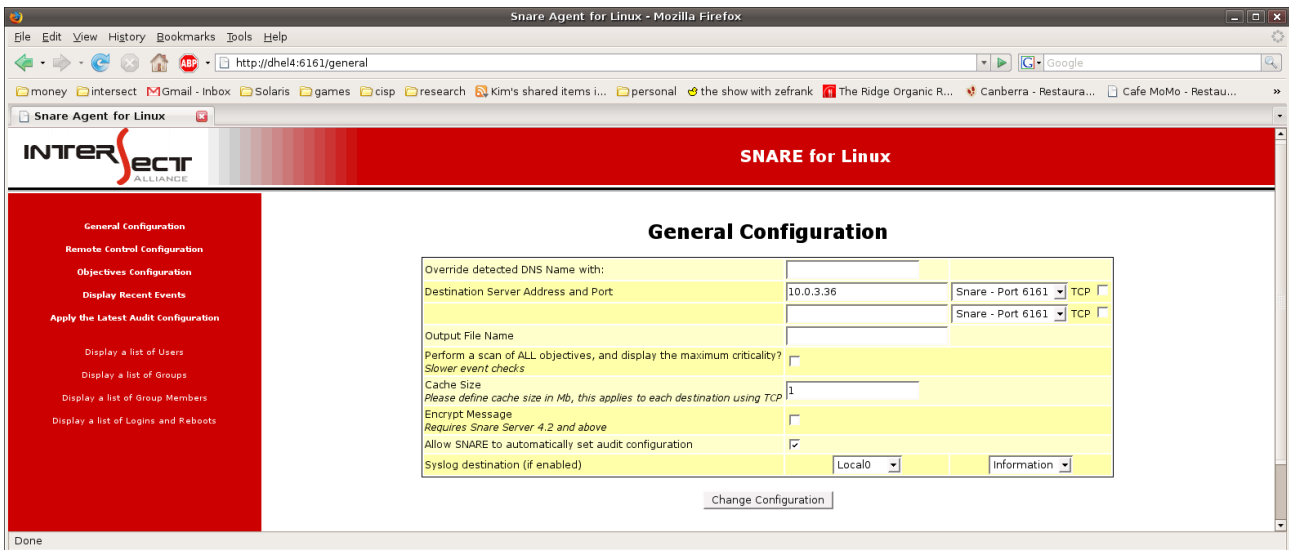


Figure 1: General Configuration

## 4.3 REMOTE CONTROL CONFIGURATION

The Snare for Linux agent, can be controlled remotely by administrators, if required.

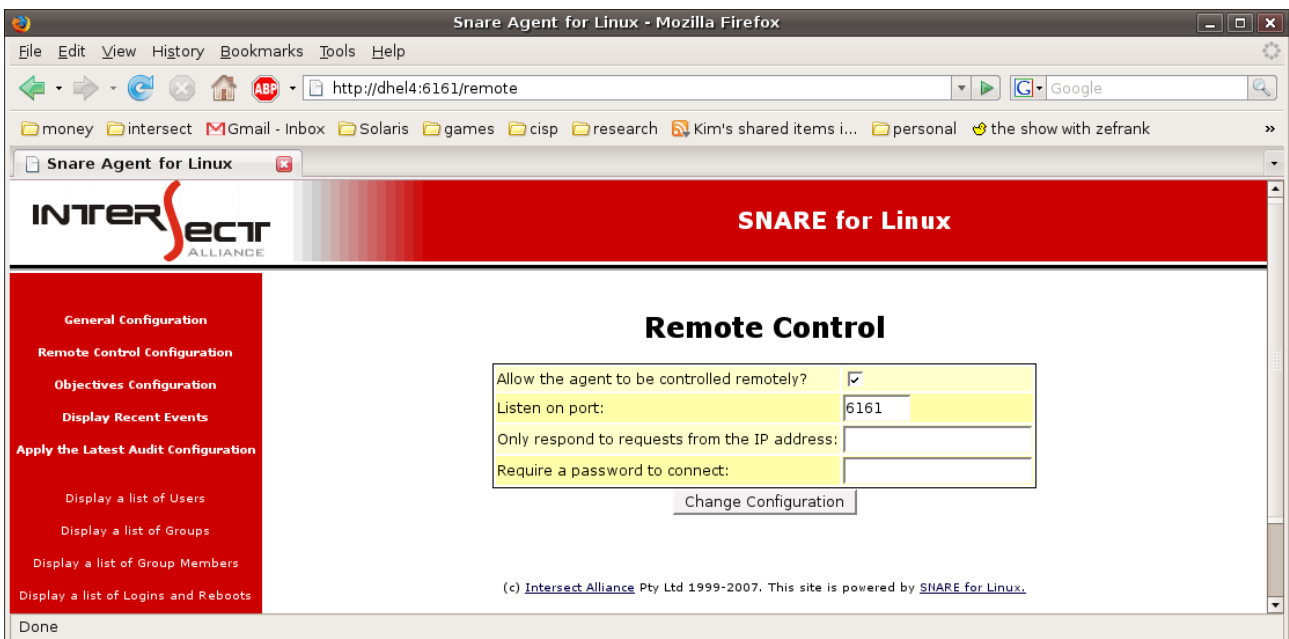


Figure 2: Remote Control

Remote control is off by default, but can be switched back on again by modifying the file `/etc/snare.conf` as per instructions in section 4.1 of this guide. An optional port that the Remote Control Interface listens on, can be specified. Users of the Snare Server should generally leave this as 6161, in order to take advantage of the Snare Server's user and group audit capabilities.

Application-level firewall capabilities are also available, which block users from accessing the Remote Control Interface from any IP address other than the one specified.

A password of appropriate strength can, and should, also be set for the remote control facility.

## 4.4 OBJECTIVE CONFIGURATION

The term 'objective' is used within Snare Agents to describe an 'auditing goal'. It is generally made up of:

- A criticality, which is sent by Snare, with the associated matching events.

The criticality levels are Critical, Priority, Warning, Information and Clear. These security levels are provided to enable the Snare user to map audit events to their most pressing business security objectives. If the criticality is set to Drop, any matching events will not be sent.

- The events that Snare should watch for.

Each of the objectives provides a high level of control over which events are selected and reported. Events are selected from a group of high level requirements, and further refined using selected filters. Events are generally grouped into the following:

- Read/Write a file or directory
  - open, creat, link, symlink, truncate, ftruncate, mknod, rename, truncate64, ftruncate64, access
- Remove a file or directory
  - unlink, rmdir
- Start/Stop program
  - execve
- Modify file/directory attributes
  - chmod, chown, lchown, fchmod, fchown, fchown32, lchown32, chown32
- Administrative events
  - mount, umount, umount2, settimeofday, swapon, swapoff, reboot, setdomainname, create\_module, delete\_module, quotactl
- Network socketcall events
  - socketcall
- Authentication events
  - login\_auth\*, login\_start\*, logout\*
- Account administration events
  - acct\_mgmt\*

In addition, any event that can be generated by the Linux audit subsystem can be specified (comma separated) by using the 'Other Events' high level group. Please note, events marked with an asterisk (\*) are not syscall events.

**Tip: Turning on file-related events can produce a very high volume of audit events on some systems, and therefore result in a considerable amount of CPU time being used by Snare and the audit subsystem.**

- A return code (generally, Success, Failure, or 'Any').

- A user match term, which may include user inclusions or exclusions.
- One or more general match terms or watches, containing a 'token' which appears within the events of interest, and the search criteria that Snare should use to include or exclude the event.

If we use the following event as an example:

```
cent4.intersectalliance.com LinuxKAudit criticality,3 event,execve,20080613 16:06:29
uid,0,root gid,0,root euid,0,root egid,0,root process,2971,ls
return,0,yes a0,8775a68 a1,875ec80 a2,8759448 a3,875ec80
arch,40000003 auid,40000003 cwd,/var/log/audit dev,fd:00 dev:1,fd:00
exe,/bin/ls flags,101 flags:1,101 fsgid,0,root fsuid,0,root
inode,97968 inode:1,146913 items,2 mode,0100755
mode:1,0100755 name,/bin/ls ogid,0,root ogid:1,0,root oid,0,root
oid:1,0,root rdev,00:00 rdev:1,00:00 sgid,0,root suid,0,root
```

Tokens are highlighted in red, and values are highlighted in blue. As such, a match term of 'mode' could be specified, with a match value of '0100755' in order to select the event above. In addition, basic wildcard pattern matching is supported, so each of the following examples would match the event:

Token	Match
euid	*,root
mode	*755
mode	?1??755
name	/bin/ls
name	/bin/*
name	*/ls
process	*,ls

You can specify that an objective needs to match more than one general match term by clicking on the green 'Add a new match term' button. The entire objective will only evaluate as a 'match', if ALL general match terms match correctly.

The order of the objectives is important if Snare is not scanning for the highest criticality. Objectives are processed from top to bottom and can be reordered from the Remote Control Interface using the arrows in the last column. An event is sent or rejected as soon as a match is found, therefore, more frequent events should be matched first. If Snare is scanning for the highest criticality, all objectives will be processed and the highest criticality will be used unless a matched objective is set to "Drop", in which case the event is rejected.

The Remote Control Interface also offers you pre-generated configurations to meet common regulatory security requirements, such as NISPOM chapter 8, Sarbanes-Oxley, or the Payment Card Industry security requirements.

Once the above settings have been finalized, the configuration may be saved to the configuration file, via the 'Change Configuration' button. However, to ensure the audit daemon has received the new configuration, the 'Apply the Latest Audit Configuration' link should be selected.

## WATCHES

Snare for Linux has two ways of auditing file-related events - event filters, and file watches. Either, or both, can be employed, depending on your requirements.

The first method (event filters) use the normal Snare objective definition process discussed above. Once the objective is activated, Snare will ask the Linux kernel to report on the appropriate events (e.g. the 'open' system call). The Linux kernel will pass EVERY 'open' call on the system back to Snare - regardless of who generates it, or which file is being opened. Snare will then use its filters to include or exclude that event according to the defined match terms. This method is extremely flexible, as you can specify wildcard matches for file names anywhere on the file system (e.g. `*/secretstuff/*`).

The second method (file watches) is somewhat different. Rather than asking the kernel to report on all file activity, a 'file watch' will cause Snare to ask the kernel to 'tag' certain files, or directories, and only generate file-related events when activity associated with those particular files occurs. This generally results in a spectacular drop in resource usage by the Snare and audit processes, as potentially thousands of file-related events-per-second no longer have to be discarded when they do not match a Snare agent objective. This method requires that each targeted file or directory must exist prior to Snare starting up. Where a directory is specified, Snare will also watch for the creation of new files and directories, dynamically including them in the set of watched files. Wildcard matches (such as `*/secretstuff/*`) will still work using file watches, but only if they occur within the 'watched' directory.

The token "watch" is used to specify a file or directory that Snare should watch (or ignore). A positive watch (=) will monitor the given file name or directory (including all subfiles and subdirectories) as long as it exists when Snare is loaded. A negative watch (!=) will ensure that the specified file or directory is ignored, even if it does not exist when Snare is loaded (e.g. in order to ignore temporary files).

In Figure 3 below, the first objective will watch the directory tree `/etc` for read/write file events, while ignoring events relating to `/etc/ld.so.cache` and `/etc/ld.so.preload` (a transient file).

The screenshot shows the 'Objective Control' configuration page in the Snare Agent for Linux web interface. The interface is displayed in a Mozilla Firefox browser window. The page title is 'Snare Agent for Linux - Mozilla Firefox' and the URL is 'http://dhel4:6161/objective'. The page features a red header with the 'INTERSECT ALLIANCE' logo and 'SNARE for Linux' text. A sidebar on the left contains navigation links for 'General Configuration', 'Remote Control Configuration', and 'Objectives Configuration'. The main content area is titled 'Objective Control' and contains a table with the following columns: Criticality, Events, Return, User, Match, Add/Remove, and Order. The table lists several objectives with their respective configurations. Below the table are buttons for 'Add a new Objective' and 'Change Configuration', and links to generate configurations for NISPOM, Sarbanes/Oxley, and Payment Card Industry.

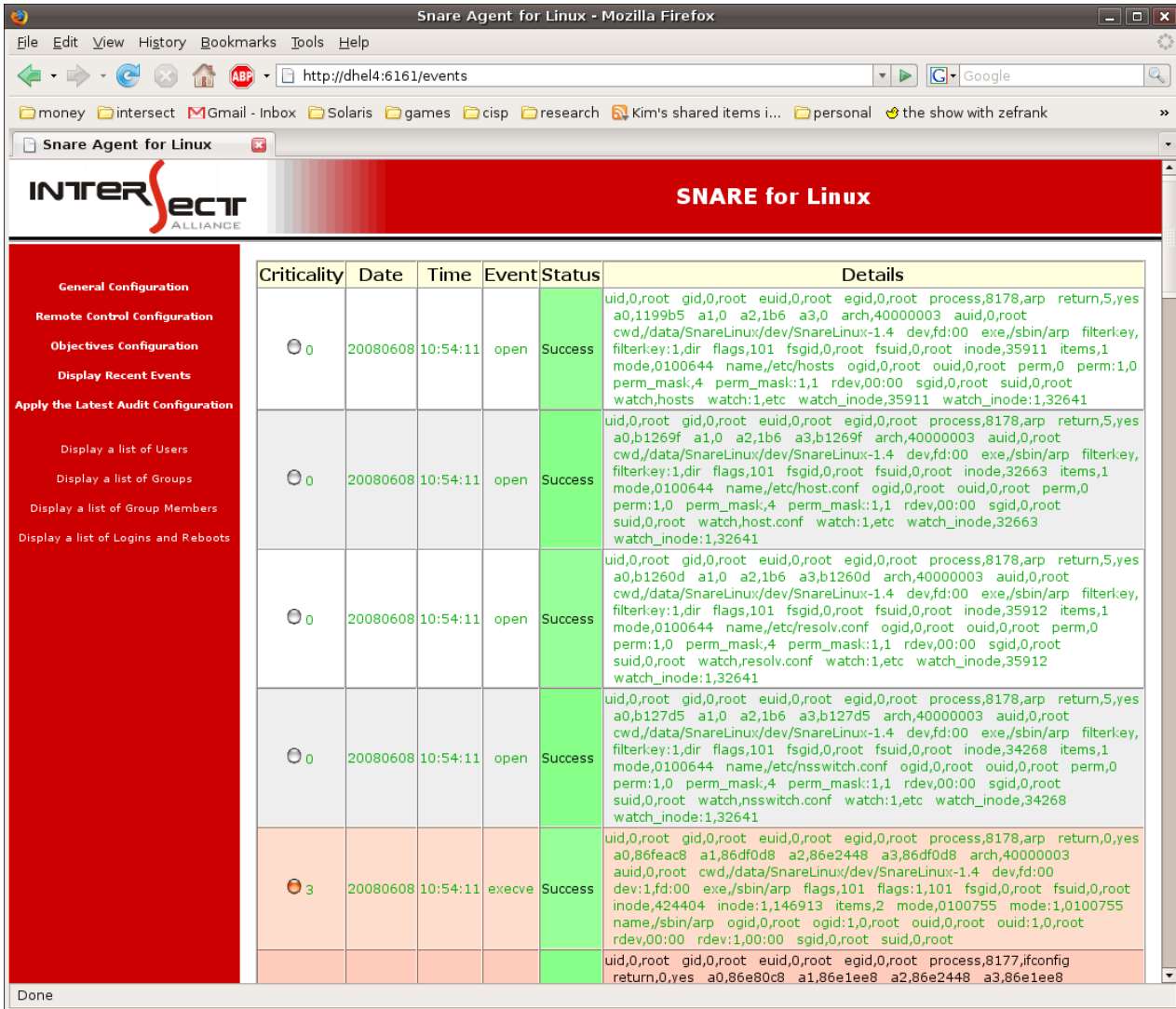
Criticality	Events	Return	User	Match	Add / Remove	Order
0 - clear	Read/Write a File/Directory open,creat,link,symlink,truncate,fr	Any	Any User	watch != /etc/ld.so.cache watch = /etc/ watch != /etc/ld.so.preload	Remove this Objective Add a new Match term	
3 - orange	Start/Stop Program execve	Any	Include Users root	=	Remove this Objective	
3 - orange	Administrative Events mount,umount,umount2,settimeof	Any	Include Users root	=	Remove this Objective	
0 - clear	Start/Stop Program execve	Any	Any User	exe = *passwd*	Remove this Objective Add a new Match term	
0 - clear	Authentication Events login_auth,login_start,logout	Any	Any User	=	Remove this Objective	
0 - clear	Account Administration Events acct_mgmt	Any	Any User	=	Remove this Objective	

[Add a new Objective](#)  
[Change Configuration](#)  
[Generate NISPOM Configuration](#) [Generate Sarbanes/Oxley Configuration](#) [Generate Payment Card Industry Configuration](#)

Figure 3: Objective Control

## 4.5 EVENT DISPLAY

A small rotating cache of audit events is kept by the Snare for Linux web server. Clicking on the 'Display Recent Events' link on the left hand side of the Remote Control Interface, will display 100 of the most recent events.



The screenshot shows a web browser window titled "Snare Agent for Linux - Mozilla Firefox" with the URL "http://dhel4:6161/events". The page header includes the INTERSECT ALLIANCE logo and "SNARE for Linux". On the left, a red sidebar contains navigation links: "General Configuration", "Remote Control Configuration", "Objectives Configuration", "Display Recent Events", "Apply the Latest Audit Configuration", "Display a list of Users", "Display a list of Groups", "Display a list of Group Members", and "Display a list of Logins and Reboots". The main content area displays a table of events.

Criticality	Date	Time	Event	Status	Details
0	20080608	10:54:11	open	Success	uid,0,root gid,0,root euid,0,root egid,0,root process,8178,arp return,5,yes a0,1199b5 a1,0 a2,1b6 a3,0 arch,40000003 auid,0,root cwd,/data/SnareLinux/dev/SnareLinux-1.4 dev,fd:00 exe,/sbin/arp filterkey, filterkey:1,dir flags,101 fsgid,0,root fsuid,0,root inode,35911 items,1 mode,0100644 name,/etc/hosts ogid,0,root ouid,0,root perm,0 perm:1,0 perm_mask,4 perm_mask:1,1 rdev,00:00 sgid,0,root suid,0,root watch,hosts watch:1,etc watch_inode,35911 watch_inode:1,32641
0	20080608	10:54:11	open	Success	uid,0,root gid,0,root euid,0,root egid,0,root process,8178,arp return,5,yes a0,b1269f a1,0 a2,1b6 a3,b1269f arch,40000003 auid,0,root cwd,/data/SnareLinux/dev/SnareLinux-1.4 dev,fd:00 exe,/sbin/arp filterkey, filterkey:1,dir flags,101 fsgid,0,root fsuid,0,root inode,32663 items,1 mode,0100644 name,/etc/host.conf ogid,0,root ouid,0,root perm,0 perm:1,0 perm_mask,4 perm_mask:1,1 rdev,00:00 sgid,0,root suid,0,root watch,host.conf watch:1,etc watch_inode,32663 watch_inode:1,32641
0	20080608	10:54:11	open	Success	uid,0,root gid,0,root euid,0,root egid,0,root process,8178,arp return,5,yes a0,b1260d a1,0 a2,1b6 a3,b1260d arch,40000003 auid,0,root cwd,/data/SnareLinux/dev/SnareLinux-1.4 dev,fd:00 exe,/sbin/arp filterkey, filterkey:1,dir flags,101 fsgid,0,root fsuid,0,root inode,35912 items,1 mode,0100644 name,/etc/resolv.conf ogid,0,root ouid,0,root perm,0 perm:1,0 perm_mask,4 perm_mask:1,1 rdev,00:00 sgid,0,root suid,0,root watch,resolv.conf watch:1,etc watch_inode,35912 watch_inode:1,32641
0	20080608	10:54:11	open	Success	uid,0,root gid,0,root euid,0,root egid,0,root process,8178,arp return,5,yes a0,b127d5 a1,0 a2,1b6 a3,b127d5 arch,40000003 auid,0,root cwd,/data/SnareLinux/dev/SnareLinux-1.4 dev,fd:00 exe,/sbin/arp filterkey, filterkey:1,dir flags,101 fsgid,0,root fsuid,0,root inode,34268 items,1 mode,0100644 name,/etc/nsswitch.conf ogid,0,root ouid,0,root perm,0 perm:1,0 perm_mask,4 perm_mask:1,1 rdev,00:00 sgid,0,root suid,0,root watch,nsswitch.conf watch:1,etc watch_inode,34268 watch_inode:1,32641
3	20080608	10:54:11	execve	Success	uid,0,root gid,0,root euid,0,root egid,0,root process,8178,arp return,0,yes a0,86feac8 a1,86fd08 a2,86e2448 a3,86fd08 arch,40000003 auid,0,root cwd,/data/SnareLinux/dev/SnareLinux-1.4 dev,fd:00 dev:1,fd:00 exe,/sbin/arp flags,101 flags:1,101 fsgid,0,root fsuid,0,root inode,424404 inode:1,146913 items,2 mode,0100755 mode:1,0100755 name,/sbin/arp ogid,0,root ogid:1,0,root ouid,0,root ouid:1,0,root rdev,00:00 rdev:1,00:00 sgid,0,root suid,0,root
					uid,0,root gid,0,root euid,0,root egid,0,root process,8177,ifconfig return,0,yes a0,86e80c8 a1,86e1ee8 a2,86e2448 a3,86e1ee8

Figure 4: View Recent Events

## 5 SNARE SERVER

The team at Intersect Alliance have produced software that works hand-in-hand with our Snare Agent software to meet organizational auditing needs. This software is known as the Snare Server.

The Snare Server is an Enterprise Audit Event Log analysis solution, comprising a central audit event collection, analysis, reporting and archive service, and security 'agents' for multiple operating systems and applications. The Snare Server software can collect data from a range of sources, including all Snare Agents, plus other sources such as network devices, firewalls, databases, and application servers.

Full source code and documentation is provided with this service, allowing the Intersect Alliance partners, or internal security professionals, to quickly develop Snare security objectives that are derived directly from your key organizational risk items. The Snare Server also comes equipped by default with an array of security objectives that allows agencies to meet common security goals. A selected screen shot (Figure 5) of the Snare Server is shown below. Full details on the Snare Server, including more screen shots are available from the Intersect Alliance web site.

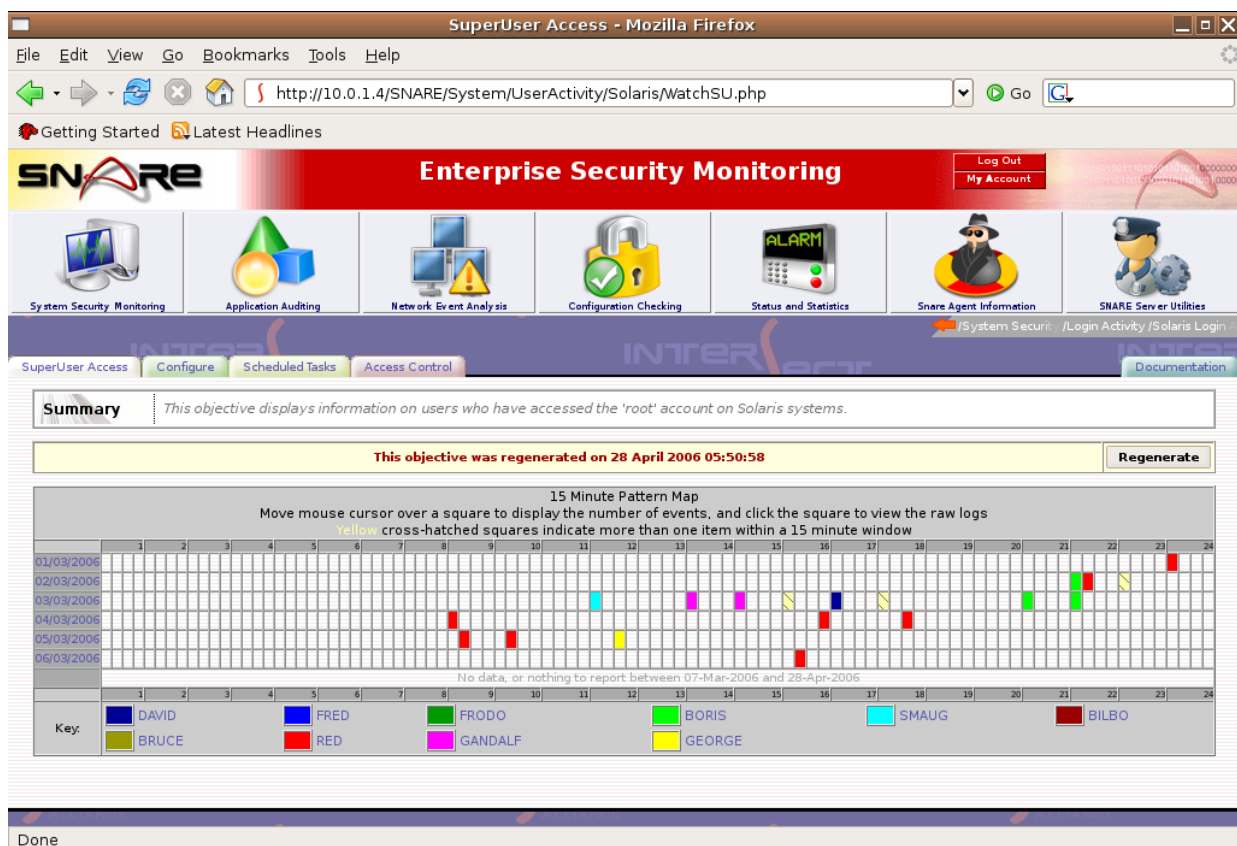


Figure 5: Screen shot from the Snare Server

## 6 ABOUT INTERSECT ALLIANCE



InterSect Alliance is a team of leading information technology security specialists in both the 'technical' and 'policy' areas. In particular, Intersect Alliance are noted leaders in key aspects of IT Security, including host intrusion detection. Our solutions have and continue to be used in the most sensitive areas of Government and business sectors. Intersect Alliance consult and contract to a number of agencies in Australia and the Asia Pacific, for both the business and Government sectors.

The Intersect Alliance business strategy includes demonstrating our commitment and expertise in IT security by releasing Open Source products such as Snare. Intersect Alliance intend to continue releasing tools that enable users, administrators and clients worldwide to achieve a greater level of productivity and effectiveness in the area of IT Security, by simplifying, abstracting and/or solving complex security problems.

Visit the Intersect Alliance website for more information at [www.intersectalliance.com](http://www.intersectalliance.com).

## APPENDIX A - CONFIGURATION FILE DESCRIPTION

Details of the audit configuration are discussed in the section on Audit Configuration. The purpose of this section is to discuss the makeup of the configuration file. The Snare configuration file is located at `/etc/snare.conf`, and this location may not be changed. If the configuration file does not exist, the audit daemon will not actively audit events until a correctly formatted configuration file is present.

Snare can be configured in several different ways, namely:

- a. Via the embedded web server (*Recommended*), or
- b. By manually editing the configuration file.

The format of the audit configuration file is discussed below.

[Remote]	This section allows you to specify settings relating to the Remote Control Interface used to control Snare.
allow=[1 0]	Turn the Remote Control Interface on or off.
listen_port=6161	Set a port that the Snare for Linux agent should listen on.
accesskey=md5password	Md5 checksum of the password used to protect the embedded web server
restrict_ip=1.2.3.4	IP address of a system that is used to remotely control the Snare for Linux agent. All requests from other systems will be dropped.

[Output]	By default, if no output section exists within the configuration file, the audit daemon will send audit data out to standard out (STDOUT). Note that audit events will be sent to all valid destinations specified in the Output section. As such, events can be sent to one or all of a file, standard output and to a remote network destination (For more than one file, and one network destination, please see the supported ).
file=/fully/qualified/file/name	The audit daemon will send data to the fully qualified filename specified within the [Output] section. Note that if the audit daemon is not running as root, the file must be writable by the user under which the audit daemon is running.
file=stdout	If stdout is specifically defined within the Output section, the audit daemon will send data to standard out.
network=hostname:port:protocol	Audit data can be sent to a remote system using the UDP or TCP protocol. Data will be sent to the remote host, and network port specified here. Protocol is only available in the supported version.

[Config]	This section allows you to specify settings relating to the operation of the Snare agent.
use_criticality=[1 0]	If enabled (1), Snare will scan all the objectives for the maximum criticality.
use_regex=[1 0]	This value determines if regular expressions should be used instead of the default wildcard matching.
clientname=override	The hostname of the client. If no hostname is set, the value of "hostname --fqdn" will be used.
set_audit=[1 0]	This value determines if Snare should set the auditing rules for the local machine.
syslog_facility=facility	The SYSLOG facility used when sending to a SYSLOG server.
syslog_priority=priority	The SYSLOG priority used when sending to a SYSLOG server.
encrypt_msg=[1 0]	This value determines if Snare should encrypt outgoing messages. This feature only appears in supported agents.
cache_size=(0 - 1024)	This value determines the size of the event cache that Snare should keep for each TCP host. The value must be between 0 and 1024. This feature only appears in supported agents.

<p>[Objectives]</p>	<p>This section describes the format of the objectives. Objectives are composed of:</p> <ol style="list-style-type: none"> <li>1. Criticality - an integer between 0 and 4 that indicates the severity of the event. 0 is 'clear', 4 is "critical". Any integer less than 0 will cause the event to be dropped.</li> <li>2. The event ID - this must either correspond to a valid auditable event, or a series of events separated by commas, and surrounded with round brackets (). Note that the embedded web server will convert the generic "groups" in the Audit Configuration window to the required events. For example, the abstracted group 'Administrative Events', will result in the event entry 'event=(mount,umount,umount2,setttimeofday,swapon,swapoff,reboot,setdomainname,create_module,delete_module,quotactl)' being written.</li> <li>3. The return code defines whether to report event (system call) if it is a success, failure or either ("**")</li> <li>4. The user list is used to audit events for selected users. Multiple users can be specified separated by commas, surrounded by round brackets.</li> <li>5. The match term(s) is the filter expression, and can be defined in extended regular expression format or standard wildcard format.</li> </ol> <p>Note that whitespace will be trimmed from the start and end of items, but will be assumed to be valid when bracketed by other characters.</p>
<p>criticality=1 event=execve return=* user=(red,george) exe=/sbin/audit*</p>	<p>Report at criticality level 1, whenever the users 'red' or 'george', attempt to execute a binary within /sbin that starts with 'audit'.</p>

## APPENDIX B - EVENT OUTPUT FORMAT

The Snare dispatcher receives data from the native Linux audit subsystem.

The native audit daemon reports data in such a way that:

- It is programatically difficult to determine how many 'lines' make up an audit event. Some lines can be repeated, with slightly different values.
- You can have multiple, identical tokens for an event (e.g. two path=)
- Event lines may be interleaved (i.e. you might get two lines from event # 1000, then one line from event # 1001, then another line from event # 1000).
- Some filename characters are translated into their HEX equivalents which will make matching filenames difficult.

A sample series of audit log lines from the native audit daemon is as follows:

```
type=SYSCALL msg=audit(1163765786.225:176189): arch=40000003 syscall=11
success=yes exit=0 a0=8d40b78 a1=8d40cb0 a2=8d28ff0 a3=8d40cc0 items=2
pid=2519 audit=0 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0
comm="w" exe="/usr/bin/w"

type=CWD msg=audit(1163765786.225:176189): cwd="/var/log/audit"

type=PATH msg=audit(1163765786.225:176189): item=0 name="/usr/bin/w" flags=101
inode=1351122 dev=fd:00 mode=0100555 ouid=0 ogid=0 rdev=00:00

type=PATH msg=audit(1163765786.225:176189): item=1 flags=101 inode=1403596
dev=fd:00 mode=0100755 ouid=0 ogid=0 rdev=00:00
```

Snare for Linux uses an internal cache to amalgamate all lines relating to an individual event, into "one line per event" format, once appropriate filtering/event selection has taken place. An event similar to the one above, will look like this once processed by Snare:

```
fc5 LinuxKAudit criticality,1 event,execve,20061117 23:16:26 uid,
0,root gid,0,root euid,0,root egid,0,root process,2519,w return,0,yes
a0,8d40b78 a1,8d40cb0 a2,8d28ff0 a3,8d40cc0 arch,40000003 audit,
0,root cwd,/var/log/audit dev,fd:00 dev:1,fd:00 exe,/usr/bin/w
flags,101 flags:1,101 fsgid,0,root fsuid,0,root
```

The Snare for Linux output format is similar to that generated by Snare for Linux 0.9.8, and is also reasonably similar to the Solaris BSM output format. Snare for Linux presents the information in a series of token/data groups. Three different field separators are used in order to facilitate follow-on processing - TABS separate 'tokens', COMMAS separate data within each token, and SPACES separate elements within data.

A 'token' is a group of related data, comprising a 'header', and a series of comma separated fields which make up data that relates to the header.

Examples of tokens:

- exe,/usr/bin/w
- return,0,yes
- euid,0,root