User Documentation

ihost

ihost is a Perl-based host for use with the i-scream Distributed Central Monitoring System. This document provides a guide to using the ihost on your Unix or Linux system.

Revision History

10/03/01	Initial creation				
		Committed by:	pjm2	Verified by:	tdb1
				Date:	24/03/01
24/03/01	Added section on firewalls				
		Committed by:	tdb1	Verified by:	pjm2
				Date:	24/03/01
24/03/01	Added section on using ihost with cron				
		Committed by:	tdb1	Verified by:	pjm2
				Date:	24/03/01
		Committed by:		Verified by:	
				Date:	
		Committed by:		Verified by:	
				Date:	

Introduction 2
What is ihost? 3
How can I get ihost? 3
How do I use ihost? 3
What does all the output mean? 4
How can I find out the address of the filter manager? 4
Running ihost from cron 4
I'm concerned about security! 5
Can I alter ihost to suit my own needs? 5
Using ihost with firewalls 5
Further information 6

Introduction

The ihost program plays a key part in monitoring systems that are monitored by the i-scream Distributed Central Monitoring System. Running an ihost on your Unix or Linux system enables machine statistics to be sent to the i-scream monitoring system.

What is ihost?

ihost is a “host” application for the i-scream Distributed Central Monitoring System. The job of a host is to harvest data from the machine on which it is running. The ihost performs this task on Solaris, Linux and FreeBSD and is written in the Perl programming language.

An ihost is self-configuring. It only needs to know the address of a single machine (called a *filter manager*), which it will connect to, obtain its configuration and then proceed to communicate periodically with another machine called a *filter*.

How can I get ihost?

The latest build of ihost may be downloaded from the *Builds* section of the i-scream project website: -

http://www.i-scream.org.uk/builds/

The website also contains other information that you may find useful in setting up an i-scream monitoring system.

How do I use ihost?

Running the ihost is usually a fairly straightforward task. Once you have downloaded the archive from the URL above, you may proceed to execute the program. This is a fairly simple process, as most of the configuration of the ihost is obtained dynamically from the filter manager.

The main program is called *ihost.pl* and can be executed in the following way: -

```
./ihost.pl <i-scream_filtermanager> <port_number>
```

The above command will start the ihost, which will then proceed to configure itself from the filter manager. The value `<i-scream_filtermanager>` should be replaced with the name of the machine that is running an i-scream filter manager and `<port_number>` should be replaced by the port number on which the filter manager is running. For example, if you were running a filter manager on a machine called raptor.ukc.ac.uk and that filter manager accepted connections on port 4567, then you would run an ihost by executing the following command: -

```
./ihost.pl raptor.ukc.ac.uk 4567
```

After an ihost has successfully performed its initial configuration, it will send all further communications to a filter machine. This may or may not be the same machine as the filter manager. The filter manager is responsible for telling the ihost which filter to use, so do not be alarmed if it suddenly changes machine.

If you receive error messages about `IO::Socket` or `Sys::Hostname` being missing, then you will need to ensure that you have these modules installed. They are part of a typical Perl installation, so the ihost should have no problems running on most Unix or Linux systems that have Perl installed.

What does all the output mean?

When you start the ihost, it will attempt to obtain its configuration from the filter manager. Progress through this stage is echoed to the screen for your information. It will display the information received from the filter manager as it receives it. This output includes details such as the address of the filter machine that the ihost will use.

Here is a typical display of an ihost during startup: -

```
% ./ihost.pl raptor.ukc.ac.uk 4567
Config started okay.
Config last modified: Sun Mar 11 01:32:45 2001
File list obtained: system.conf;computing.conf;
FQDN returned: raptor.ukc.ac.uk
UDP packet period: 10 seconds.
TCP heartbeat period: 60 seconds.
Config ended.
Got filter data (raptor.ukc.ac.uk, 4589, 4589)
Host successfully configured via TCP.
Configuration finished sucessfully!
```

When an ihost is functioning normally, it will then proceed to display output similar to this: -

```
------^------^------^------^------^--- etc
```

Each - indicates that a UDP packet has been sent to the filter. UDP is a type of network protocol, and these packets contain information about the machine that the ihost is running on.

Each ^ indicates that a heartbeat has been sent to the filter using TCP. TCP is another type of network protocol, but is more reliable than UDP. Heartbeats are sent to ensure that the filter still knows about the existence of the ihost. Heartbeats are typically configured to be sent less often than UDP packets.

If the configuration of the i-scream server changes, you may see the host reconfigure itself (with similar output to that seen when the program is started). This is normal behaviour and enables the ihost to be dynamically updated without human intervention.

How can I find out the address of the filter manager?

The filter manager is the section of the server that handles ihost configuration and assignment to filters. You will need to know the hostname and port number of the filter manager machine before you are able to use the ihost. If you have trouble finding out this information, then please contact the person(s) responsible for setting up the i-scream filter manager on your network.

Running ihost from cron

Often you want to leave ihost running unattended, and want to be sure it will restart if stopped for any reason (crash, reboot, etc). To enable you to do this there is a script named `ihostchk.sh` which can be run from a cron script. Run periodically it will check whether ihost is running, and restart it if for any reason it's not found.

A few minor modifications need to be made to the script before it will work. These are listed in the first few lines of the script, and have explanatory comments. They tailor the script to suit your needs and environment.

Once this is done you simply add a line to your crontab (`man crontab` for details). The following line checks ihost every 10 minutes.

```
0,10,20,30,40,50 * * * *      /path/to/ihostchk.sh
```

This will result in an e-mail being sent to the user (who owns the crontab) every time an ihost is restarted, so you may like to modify it as follows if you don't want these e-mails.

```
0,10,20,30,40,50 * * * *      /path/to/ihostchk.sh >/dev/null 2>&1
```

In most situations the latter is best, but if you want to be kept informed of errors and restarts, use the former. Left to it's own devices this script will keep ihost running indefinitely.

I'm concerned about security!

Although we have not taken (yet) any steps to make the entire i-scream system secure, we are very confident that running ihost does not pose a risk. ihost only sends data and does not allow any users to connect to it. It is unlikely that ihost will send any data that may be considered highly sensitive, however, if you are concerned about this, then you may like to review the documentation to find out exactly what is sent.

Can I alter ihost to suit my own needs?

Yes, of course you may. ihost has been written in Perl so that people with a basic knowledge of the language (or, indeed, any similar language) will be able to have a good go at customising the way in which it works.

The ihost consists of two files: -

1. statgrab.pl – responsible for collecting the data from the system.
2. ihost.pl – responsible for configuration, XML data formatting and sending.

If you are proficient in Perl, then you could start by altering statgrab.pl to read new data from your system. Once this is done and tested, you could modify ihost.pl to include this data in outgoing packets.

A more in-depth discussion of altering ihost.pl may be found in the maintenance documentation for ihost.

Using ihost with firewalls

A firewall is commonly used in networked environments to protect the computers from external intrusion. This is often done through restricting the data that can flow between parts of the network. If this applies to you, please ensure you read this section so you are aware of what ihost attempts to send.

As has been previously discussed, ihost begins by communicating with the filter manager section of the server. This is done by making a single TCP connection, to a defined port, from ihost to the filter manager. No connections are ever made back to ihost from the filter manager.

During the general running of ihost, communication is made to a filter. The hostname and port of this machine are not usually defined, and are configured automatically by the filter manager. However, the person(s) running the filter will be aware of the hostname and port, and will be able to provide these details.

Communication with the filter is in two parts. The first part is sending UDP packets of data to the filter. These are sent on a regular basis and are not usually large. No UDP traffic is ever sent back to the ihost. There is also a TCP 'heartbeat' communication sent on a less regular basis. This is always established by ihost. It is possible that the filter will be configured to run a series of service checks on a host machine. These are always initiated in response to a 'heartbeat' communication, and are of course configurable on the server side.

In summary, ihost needs to be able to establish TCP connections to both the filter manager and the filter. It also needs to be able to send UDP packets to the filter. If configured to do so, the filter must be able to connect to configured service ports (eg. http, ftp, smtp) on the host, although this is not essential for normal operation.

Further information

Further information is available in our other documentation; the latest versions of which may be found online at the project website. Thank you for using i-scream products.

http://www.i-scream.org.uk