Host to Filter protocol (XML)

Expected and Recommended data from Hosts

This document is intended to provide third parties with the knowledge required to use their own implementation of a piece of host monitoring software. In particular, this document details the data that is to be expected from a Host. This document is additional to the recommended way of storing XML in UDP packets.

Revision History

07/03/01	Initial creation				
		Committed by:	pjm2	Verified by:	tdb1
				Date:	24/03/01
		Committed by:		Verified by:	
				Date:	
		Committed by:		Verified by:	
				Date:	
		Committed by:		Verified by:	
				Date:	
		Committed by:		Verified by:	
				Date:	

Introduction

This document is intended to provide third parties with the knowledge required to use their own implementation of a piece of host monitoring software. In particular, this document details the data that is to be expected from a Host. This document is additional to the recommended way of storing XML in UDP packets.

Background

Hosts are expected to periodically send UDP packets to the central monitoring system. Such packets may contain various pieces of information about the host, such as how much free memory is remaining on the host, etc. Some of this data is deemed to be 'essential', some is 'expected' and the rest is not necessary, but shall not be ignored by the central monitoring system server.

Specification

The UDP packet must contain a complete and well-formed piece of XML mark-up, describing the data that the host is submitting. In the event of a piece of data containing an invalid character, such as a "<", it is the **responsibility of the host** to format this in a valid manner, for example by replacing all occurrences of "<" with "&lt" before it is sent in a UDP packet to the central monitoring system server.

Alternatively, the data may be wrapped up in special CDATA tags. For example, if we wish to send the following text:

```
<>&!"
```

surrounded by `<test_chars>` tags then we can safely send this by wrapping it up to produce the following XML snippet:

```
<test_chars><![CDATA[<>&!"]]></test_chars>
```

The bare minimum that a host should send is the following: -

```
<packet>
</packet>
```

Note that every XML tag must also have a matching closing tag.

However, such a packet is rather useless when it is received by the server - it does not provide any useful information. As such, a packet will be rejected by the server if it does not specify the `machine_name`, its I.P. address (`ip`), a packet sequence number (`seq_no`) and a `date`. Note that the date is a long integer representing the number of seconds since the epoch. All of these are expected to be specified as an attribute of the packet tag. If they are not specified here, then the packet will be rejected. Finally, the packet tag must contain an attribute called `type`, which *must* have the value of `data`.

Thus, to ensure that a packet has a chance of being acted upon, a host must specify at least five attribute values - `machine_name`, `ip`, `date`, `seq_no` and `type`. These must be formed in the style shown below.

```
<packet machine_name="raptor.ukc.ac.uk" ip="aaa.bbb.ccc.ddd"
        date="93456245245" seq_no="1234567" type="data">
</packet>
```

Note that the server may run additional 'plugin' tools that are designed to filter/reject packets under certain conditions. Please check with the server administrator if you wish to know about the specific configuration of your central monitoring system.

The I.P. address and machine_name could be found from the UDP packet itself, but these remain essential so as to provide resilience from stray packets that do not contain any valid or useful data.

Remember that malformed XML data would be rejected by the central monitoring system immediately, without acting upon it.

Classification of XML data in the i-scream system

ESSENTIAL

The following values are **essential** in each packet. Any packet without these will be rejected. They **must** be specified as attributes of the root node ("`packet`"): -

```
ip
seq_no
date
machine_name
type
```

EXPECTED

The following values are **expected** in each packet. All packets are **recommended** to specify these values, as the server is designed to permanently store such data in an easy and quick to access manner. It is not essential to specify such values, but it does limit the usefulness of the central monitoring system if these values are not specified: -

```
os *
cpus *
load *
memory *
swap *
disk *
users (attribute)
```

* - These parameters are specified with tags of the same name. They contain nested XML tags. See later in this document for more details about that!

All of the values marked as being attributes above must be specified as an attribute of the root tag ("`packet`")

OTHER

Any other arbitrary values may be specified by the host. The only provision here is that they may not share the same name as another tag or attribute within the same hierarchical level.

The server will be able to handle such clashes, however, it cannot be guaranteed which of the duplicate parameters would gain priority.

Example of a complete "expected" XML packet

I shall start by showing an example of a complete 'expected' packet. I shall then explain the contents of it.

Here is an example of an XML packet, containing all of the *essential* and *expected* data: -

```
<packet seq_no="123" machine_name="raptor.ukc.ac.uk"
date="93452345454" type="data" ip="1.2.3.4">
    <load>
        <load1>0.45<load1>
        <load5>0.23<load5>
        <load15>0.12</load15>
    </load>
    <os>
        <name>SunOS</name>
        <release>5.8</release>
        <platform>sun4u</platform>
        <sysname>raptor</sysname>
        <version>Generic_108528-06</version>
        <uptime>1657260</uptime>
    </os>
    <users>
        <count>3</count>
        <list>pjm2 root root</list>
    </users>
    <processes>
        <total>12</total>
        <sleeping>1</sleeping>
        <zombie>1</zombie>
        <stopped>0</stopped>
        <cpu>10</cpu>
    </processes>
    <cpu>
        <idle>98.2</idle>
        <user>1.2</user>
        <kernel>0.0</kernel>
        <iowait>0.0</iowait>
        <swap>0.7</swap>
    </cpu>
    <memory>
        <total>4096</total>
        <free>2934</free>
    </memory>
    <swap>
        <total>3020</total>
        <free>3020</free>
    </swap>
    <disk>
        <p0>
            <free>1234</free>
            <total>1234</total>
            <mounted>/</mounted>
            <label>xyz</label>
        </p0>
        <p1>
            <free>123</free>
            <total>123</total>
            <mount>/tmp</mount>
            <label>xyz2</label>
        </p1>
    </disk>
</packet>
```

Note that linefeeds and spaces would normally not be present. These are only shown above for structure clarity.

Note that all of the essential attributes have been specified as attributes of the root node. The expected attribute values are also attributes of the root node.

The `<cpu>` tag begins a list of cpu load information. The values here should be expressed as percentages, without the % symbol.

The `<load>` tag begins a list of system load information. The load tag is only expected to contain 3 sub tags, named "`load1`", "`load5`" and "`load15`". These represent the average load 1 minute ago, 5 minutes ago and 15 minutes ago respectively. This is not expected from NT systems.

The `<users>` tag begins a section containing details about the users on the system. There are two recommended sub tags within this section: -

- `<count>` - The total number of users on the machine.
- `<list>` - The list of user logins.

The `<memory>` tag begins a list of system memory information. This tag is only expected to contain two sub tags, as shown in the example above. These two sub tags specify how much memory the system has free and how much memory the system has in total. These values are in megabytes.

The `<swap>` tag is identical in nature to the `<memory>` tag.

The `<processes>` tag contains the following fields: -

- `<total>` - Total number of processes on the machine
- `<sleeping>` - Number of sleeping processes
- `<zombie>` - Number of zombie processes
- `<stopped>` - Number of stopped processes
- `<cpu>` - Number of CPU processes

The <disk> tag begins a list of partition information. Each sub tag must be of the form px, where x is the number of the disk partition that is unique for that particular system. Note that the numbering of px is expected to start from zero. Each <px> tag contains the following sub tags: -

- `<free>` - the free partition space in bytes.
- `<total>` - the total partition size in bytes.
- `<mount>` - the mount point, e.g., "/var".
- `<label>` - the label of the physical disk.

Any other values and attributes may be specified in the packet, providing they are not specified in the same hierarchical level as another attribute or value with the same name.

Flexibility and minimising bandwidth

Please note that the following issues still stand: -

The means of submitting host data via UDP containing XML mark-up is provided so that future customisation is easily possible. It would be possible to easily tailor a custom piece of host monitoring software to provide exactly what data is desired for adequate monitoring.

Some of the XML mark-up demonstrated above contains a lot of redundant features. For example, it is not necessary to lay the contents out neatly (although this certainly helps visualise the contents).

The amount of data sent within each UDP packet may be (in some cases, vastly) reduced by using some of the ideas described below: -

1. Remove unnecessary linefeeds and 'white space'
2. If a single piece of data is to be represented, it will usually occupy less space if it is stored as an attribute to a tag, rather than within a pair of tags.
3. Comments within the XML may be useful for testing purposes, however, the server ignores all comments so these can be removed to reduce packet sizes.

Taking the above into account, here is an example of some ideally formed XML to be contained within a UDP packet: -

```
<packet machine_name="raptor" ip="aaa.bbb.ccc.ddd"
><freespace><drive1>23677</drive1><drive2>23534</d
rive2><drive3>10390</drive3></freespace></packet>
```

Notice how all unnecessary 'white space' and linefeeds have been removed. The comment has also been removed. Values such as "`machine_name`" and "`ip`" have both been stored as an attribute of the root node ("`packet`") as this results in a smaller packet size.