Host to Filter protocol (XML)

Recommendations for the Host to Filter protocol

This document covers some of the suggested recommendations for laying out data in XML to be sent from a host to a filter.

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 shall help to provide a separate party with the knowledge required to use their own implementation of a piece of host monitoring software. In particular, this document details the expected manner of data transfer from a host to the central server via a filter.

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.

Specification

It is the responsibility of the host monitoring software to realise where to send its data, by means of some auto-configuration system, or otherwise.

Each discrete bundle of data from the host must be sent via a UDP packet to the central monitoring system. There is no limit to the size of this packet, however, the server may reject packets that are too large. The central monitoring system may ignore any data after the first 8kb of each packet, resulting in the possibility of such packets being rejected due to malformed/incomplete contents.

The UDP packet must contain a complete and well-formed piece of XML mark-up, describing the data that the host is submitting.

The XML contents of the UDP packet may define a document encoding standard, however, this is not a necessity as a default encoding can be assumed, this being suitable in most cases.

Any packets that do not parse as being valid XML shall be rejected by the server. This is likely to also include any packets that have the closing root tag placed after 8kb from the start of the packet's contents.

The XML markup within a packet is typically used to specify the data that the host is submitting. This must consist of at least a root tag, called "packet".

For example, 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.

The server can recognise parameter values within tags, such as: -

```
<packet machine_name="raptor" ip="aaa.bbb.ccc.ddd">
</packet>
```

Data to be transmitted may be defined within the parameters of tags (as above) or it may be defined within its own tags, vis: -

```
<packet>
    <machine_name>raptor</machine_name>
    <ip>aaa.bbb.ccc.ddd</ip>
```

```
</packet>
```

To avoid confusion, it is clearly necessary to escape any characters that may be incorrectly misinterpreted as XML by the parser.

The XML structure may be free-form. Any leading and trailing spaces are ignored in values. For example, the following defines exactly the same data as the above example: -

```
<packet>
    <machine_name>
        raptor
    </machine_name>
    <ip>
        aaa.bbb.ccc.ddd
    </ip>
</packet>
```

In cases where multiple data may be present, it may be more useful to nest tags to a number of levels. For example: -

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

Such formatting is perfectly acceptable by the server. Packets may also contain comments, for example: -

```
<packet>
    <!-- This is a comment! -->
    <machine_name>raptor</machine_name>
    <ip>aaa.bbb.ccc.ddd</ip>
    <freespace>
        <drive1>23677</drive1>
        <drive2>23534</drive2>
        <drive3>10390</drive3>
    </freespace>
</packet>
```

Remember that malformed XML data would be rejected by the central monitoring system without acting upon it. Thus, we urge would-be host developers to take care.

Flexibility and minimising bandwidth

The means of submitting host data via UDP containing XML markup 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 markup demonstrated above contains a lot of rendundant 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, this means that the final XML example above may be turned into the following without losing any information: -

```
<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.