

IP BlackLists

by SuperBonBon

1. JAFS IP Blocking

JAFS allows to block certain ranges of ip addresses using the [singleEntry](#) XML element from the [server configuration](#) or a list parser.

2. Simple IP block range

Here to block an entire range of IP addresses you simply need to add a [singleEntry](#) block to the [blackList](#) xml element in the [server configuration file](#).

Here are the attributes description :

- name: This is the name of the entry
- startIpRange: The block start ip range, needs to be an IP V4 address
- stopIpRange: The block stop ip range, needs to be an IP V4 address

Here's a small example to block all IP from 192.168.1.1 to 192.168.1.33 and 192.168.1.64 to 192.168.1.255 range :

```
<server xmlns="http://www.sbbi.net/jafs/1.0/jafs-server"
  ...
  code="sample_server" ... >
  ...
  <blackList>
    <singleEntry name="my first block"
      startIpRange="192.168.1.1"
      stopIpRange="192.168.1.33"/>
    <singleEntry name="my second block"
      startIpRange="192.168.1.64"
      stopIpRange="192.168.1.255"/>
  </blackList>
  ...
</server>
```

3. Using a list from the internet to block IP's

You can create your own lists or use existing lists from the internet and use the list entries to block access to your server quite easily if you're a programmer.

Let's assume the following scenario:

- I know a list of undesired ip's published on some website at <http://www.somesite.com/somelist.txt>
- The format of the lines in the list is as follow (a dream scenario :o): Entry Description - 192.168.1.64 - 192.168.1.255
- I want to integrate it into JAFS

Here's the solution :

First you'll need to create a Java class that implements the `net.sbbi.jafs.ipblocklists.BlackListedIPListParser` interface :

```
package net.sbbi.jafs.ipblocklists;

import java.io.*;
import java.util.*;
import org.apache.avalon.framework.logger.*;
import java.net.*;
import net.sbbi.jafs.services.*;

public class MyListParser implements BlackListedIPListParser {

    public Set parse( URL location, Logger log ) throws Exception {
        if ( location == null ) throw new Exception( "This parser needs an URL to work" );
        Set entries = new HashSet();
        // we connect to the list URL
        java.net.URLConnection conn = location.openConnection();
        java.io.InputStream input = conn.getInputStream();
        // we create the streams and readers
        // to read the list
        InputStreamReader inReader = new InputStreamReader( input );
        BufferedReader inLineReader = new BufferedReader( inReader );

        String line = null;
        // we process each line of the list
        while ( ( line = inLineReader.readLine() ) != null ) {
            // each entry match the following format :
            // Entry Description - 192.168.1.64 - 192.168.1.255
            // we just split the data using the " - " delimiter
            String[] entryData = line.split( " - " );

            if ( entryData != null && entryData.length == 3 ) {
                // ok we have a good line, we simply add the entry
                BannedIPRangeEntry entry = new BannedIPRangeEntry();
                entry.setEntryName( entryData[0] );
                entry.setStartIpRange( entryData[1] );
                entry.setStopIpRange( entryData[2] );
                try {
                    // we validate the data
                    entry.validateEntry();
                } catch ( Exception ex ) {
```

IP BlackLists

```
        log.warn( "Invalid data in line " + line +
                  " for list location " + location.toString() );
    }
    entries.add( entry );
} else {
    log.warn( "Unable to parse line " + line +
              " for list location " + location.toString() );
}
}
// we close all the readers and streams
inLineReader.close();
inReader.close();
input.close();
// and finally we return the list that contains
// all the BannedIPRangeEntry entries
return entries;
}
}
```

Note:

We recommend you to put your parser classes into the *net.sbbi.jafs.ipblocklists* package (for parsers classes auto listing via JMX interface) however this is not mandatory.

Once you are done, simply compile this class using your favorite java compiler. Don't forget to include the *jafs.jar* and *avalon.jar* libraries located in the JAFS *libs* directory in the classpath to compile it..

After compilation you'll need to zip the generated java class, rename this file to *.jar* extension and copy it in the JAFS *libs*

Last step is to configure [JAFS server file](#) :

```
<server xmlns="http://www.sbbi.net/jafs/1.0/jafs-server"
  ...
  code="sample_server" ... >
  ...
  <blackList>
    ...
    <list location="http://www.somesite.com/somelist.txt" name="MyFirstList"
      parserClassName="et.sbbi.jafs.ipblocklists.MyListParser" rescantTi
    ...
  </blackList>
  ...
</server>
```

JAFS provides a parser for the Peerguardian 1 and 2 IP block lists (*net.sbbi.jafs.ipblocklists.PeerGuardianListParser* and *net.sbbi.jafs.ipblocklists.PeerGuardian2ListParser*) which seems to be the most popular on

internet currently.