ACS v5 Log Collector/Parser Configuration Guide

Introduction

ACS5 does not log to CSV format files and does not export any logs in the same fashion as ACS4. Therefore a new set of tools are required if customers are to benefit from exporting event data in CSV format for importing into Extraxi aaa-reports! or another database.

The default configuration of this toolset will convert ACS5 format logs to CSV format with the same column headings as used by ACS v4 for import into Extraxi aaa-reports! It is possible to replace the default parser configuration file to create alternative CSV formats for use with other reporting systems or databases.

Installation

The installer file can be obtained directly from the Extraxi.com web by requesting a trial version and then following the emailed links to the download area. The installer file will be of the form acs5logs-vNNN-setup.exe. The installer should be run using the "add programs" control panel applet from an account in the local administrators group. This will ensure appropriate permissions and avoid problems if using a remote RDP/Terminal services connection.

Overview

The toolset uses the ACS5 feature of backing up its log data to an FTP server (called a repository), downloading the backup, and extracting the ACS5 logs (**GetAcs5Logs.exe**) and then processing with a parser (**AcsSysLogToCsv.exe**). The parser has the ability to generate CSVs that contain ACS5 specific columns, or in the case of customers wishing to import data into aaa-reports! to co-exist with an existing ACS4 server, can also map ACS5 data to look approximately like ACS4.

The toolset comprises several tools and scripts that together perform the following actions:

1. Connects to the ACS via SSH and initiates a backup of ACS logs which are uploaded to an FTP site
2. Download the backup via FTP for processing
3. Extract the acs logs from the uploaded backup file
4. Parse the logs into the CSV files

Folder Layout

In the installation folder there are a number of sub-folders as below:

- **Acs**. Raw ACS Logs are automatically copied to this folder prior to being parsed. The parser will move acs logs to a "processed" folder which prevents the same file being processed multiple times.
- **Bin**. Location of various script, config and .exe files
- **Csv**. CSV data created by the parser will be copied into this folder ready for importing into aaa-reports! You may either configure aaa-reports! to use as the source folder for its input or copy the files to another location.

Running the tools

To initiate the process of collecting and generating CSV data the script file *Bin\ACS5toCSV.bat* is executed either manually or via the "at" command. Note that on Windows 7/Server 2008 (or later) the "Run As Administrator" option will be required when launching the command prompt window when the user is not logged in as Administrator.

Pre-requisites

Before you start there are several configuration tasks to be performed:

1. Download and install Microsoft .NET v4 runtime (required for the parser)
2. Configure your FTP Server to receive backups uploaded from the ACS server(s)
3. Login to the ACS console and create a "repository" configuration for your FTP server
4. Edit the script files and .ini files to match your ACS Server details

The above tasks are described further in the following sections.

1. Install Microsoft .NET v4

The .net runtime may be downloaded from http://www.microsoft.com/download/en/details.aspx?id=17851

2. Configure FTP Server

The FTP server instance can reside anywhere on your network, or possibly even on this server – however there must be sufficient network connectivity for both the ACS server and the Extraxi tools to login using standard FTP authentication with both read/write permissions. For testing purposes create a README.TXT file and place it in the physical folder associated with the FTP server login.

You will require the FTP server name/ip address plus login username and password.

2. Setup ACS Repository

To create the repository configuration you will need to connect to the console of each ACS server either physically, or via remote SSH login. Note that ACS5 does not support standard telnet.

Once logged in enter the following commands:

```
configure
repository extraxi
url ftp://<ipaddr or hostame of ftp server>
user <ftp username> password plain <ftp password>
exit
```

for example:

```
configure
repository extraxi
url ftp://10.0.0.2
user acs password plain cisco
exit
```

To test the repository is configured and that there is connectivity to the chosen FTP server, enter the commands:

```
show repository extraxi
```

This will effectively run a 'dir' command on the FTP server and return the names of any files present in the physical folder that the FTP server was mapped to. If you created the README.TXT file while setting up the FTP server this should be displayed.

3. Edit Scripts & .ini files

There are several files that will require editing **prior** to execution:

Bin\ACS5toCSV.bat

Edit the ACS5toCSV.bat to add details of each ACS server that you wish to collect logs from. For each ACS add a line as follows:

```
call GetLogsFromACS <ACS Name>
```

For example:

```
call GetLogsFromACS MyACS1
call GetLogsFromACS MyACS2
```

Note. The GetLogsFromACS script file performs the bulk of the tasks and should not edited unless guided by a support representative.

Bin\GetAcs5Logs .ini files

In order to connect via SSH to your ACS servers an SSH configuration file is **required** for each ACS. Located in the bin\getAcs5Logs folder there must be a .ini file for **each** named ACS Server – the filename **must** match the name of the ACS server.

Using the previous example of two servers called MyACS1 and MyACS2 you will require two .ini files:

Bin\ getAcs5Logs\MyACS1.ini
Bin\ getAcs5Logs\MyACS2.ini

Each .ini file should contain the following values:

[SSH]
hostname=<FQDN name/*ipaddr of ACS*>
user=<*ACS console username*>
password=<*ACS console password*>
repository=<*Name of repository configured to receive backups*>
timeoutMins=<*max timeout when waiting for backup to complete*>

[FTP]
hostname=<FQDN name/*ipaddr of FTP server to which ACS uploads backups*>
user=<*FTP username*>
password=<*FTP password*>
dir=<*optional directory name if files not placed in ftp root folder*>

[TAR]
rawACSLogsFolder=<*destination folder for acs logs*>

for example:

[SSH]
host=10.0.0.1
user=admin
password=cisco
repository=extraxi
timeoutMins=10

[FTP]
host=10.0.0.2
user=acs
password=cisco
dir=

[TAR]
rawACSLogsFolder=..\..\ACS

Note that when ACS console authenticates the admin user credentials, **both** the username and password are case sensitive. Ie user *Admin* is not the same user as *admin*.

Running the tools

Once the configuration tasks (above) have been completed the process of collecting logs and generating CSV files is initiated by running the ACS5toCSV script:

```
D:\Extraxi\ACS5Logs\Bin>acs5tocsv

GETACS5LOGS v0.1.10 - ACS CSV File Remote Collector, Copyright 2011, Extraxi Ltd
Invoked with config from .\MyACS.ini
Attempting to SSH onto [MyACS] at [10.0.0.1] to request backup
ACS response: [file backup-MyACS-111206-1601.tar.gz uploaded ok]
Attempting to FTP [backup-MyACS-111206-1601.tar.gz] from [10.0.0.2]
Attempting to unpack logs from [C:\Temp\backup-MyACS-111206-1601.tar.gz] to [..\..\Acs]
Done.

Parsing ACS logs into CSV files. This may take several minutes...
Done.

D:\Extraxi\ACS5Logs\Bin>
```

Results

Provided the various stages completed without error the CSV folder should contain a set of CSV files created by parsing the raw ACS logs.

Parser Configuration

The supplied parser config file (ACS5Config.xml) will map ACS5 log data to an *approximation* of ACS4 CSV format data using the original attribute names etc. Therefore any reporting tool setup to receive ACS4 log data should mostly work. It must be stressed that ACS5 does not annotate RADIUS and TACACS+ accounting records with extra fields (eg group, real name etc) and therefore this data will not be available in the parser generated CSVs.

The parser configuration defines a series of *transformations* that map certain ACS Events to a named CSV file. In ACS4Config.xml events have been grouped to match the CSV files generated by ACS4, for example Failed Attempts, Passed Authentications, RADIUS Accounting etc. It is possible to create an entirely new config that maps ACS events to completely new CSV file names and formats.

Each CSV will include a number of fixed columns such as Date, Time and AAA Server name plus a set of optional fields parsed out of the logged event, for example as below:

```
<Transform Name="Tacacs+ Administration" Events="3300">
  <Format Type="CSV">
    <Output>..\..\CSV\TACACS Admin</Output>
    <DateFormat>dd/MM/yyyy</DateFormat>
    <Columns>
       <Column Attribute="Message.Reason"              Heading="Reason"/>
       <Column Attribute="AcctRequest-Flags"          Heading="Acct-Flags"/>
       <Column Attribute="User"                       Heading="User-Name"/>
       <Column Attribute="Device IP Address"          Heading="NAS-IP-Address"/>
       <Column Attribute="Port"                       Heading="NAS-Port"/>
       <Column Attribute="Privilege-Level"            Heading="priv-lvl"/>
       <Column Attribute="Remote-Address"             Heading="Caller-Id"/>
       <Column Attribute="Service-Argument"           Heading="service"/>
       <Column Attribute="CmdSet"                     Heading="cmd"/>
       <Column Attribute="SelectedAccessService"      Heading="User Field 4"/>
       <Column Attribute="NetworkDeviceGroups"        Heading="Network Device Group"/>
       <!-- The columns below are pulled from the AVPair wrappers for T+ protocol attrs -->
       <Column Attribute="task_id"/>
       <Column Attribute="priv-lvl"/>
    </Columns>
  </Format>
</Transform>
```

Further discussion is beyond the scope of this document; suffice to say that new ACS events and csv column may be added (or removed) as required.

Transforms

Each named transform equates to a particular CSV log target and is used to define a set of ACS5 events that we be included in the that transform, eg

```
<Transform Name="Tacacs+ Administration" Events="3300,3301">
      <Format Type="CSV">
        <Output>..\..\CSV\TACACS Admin</Output>
```

In the XML snippet above a transform for Tacacs+ administration events is defined that will include ACS5 events 3300 and 3301. The output type is CSV and the base filename is “TACACS Admin”. Note that the actual filenames generated will include the timestamp of the first event, eg “TACACS Admin 12-Dec-2011(13:20:02).CSV”. The parser can process a single ACS log or a whole set will generate a separate CSV file for each ACS log file. Also a relative file path (to the parser .exe) has been specified but any folder could be used.

Refer to Cisco ACS documentation for a complete list of ACS event numbers and their meaning.

Columns

For each transform it is possible to define the columns that will appear in the resulting CSV file. Each row in the CSV will comprise:

- Default header columns. The parser will automatically insert Date, Time and AAA Server columns
- Optional ACS event header columns. Every ACS event has a number of standard values that may optionally be included:
  - Severity. Eg NOTICE, WARN, ERROR etc
  - Class. eg Passed-Authentication, Failed-Authentication etc
  - Event. The specific ACS event number Eg 5200
  - Reason. The textual summary of the Event, eg “Authentication succeeded”
- Optional attribute columns. These vary depending on the particular message, but typically include User, Port, Device IP Address etc

Renaming Header Values and Attributes

All columns (with the exception of default headers) can be renamed by optionally including a Header clause, as in the example below:

```
      <Column Attribute="Message.Reason"              Heading="Reason"/>
```

Attribute Handling

How ACS attributes are parsed will depend on the method of encoding in the event message itself and whether they are single or multiple instance (ie how many times any attribute may appear in a single event). In this respect ACS is somewhat inconsistent with some attributed appearing multiple times, others appearing once, but having multiple values. Some prime examples of a parser-writer’s nightmares are given below:

```
      Step=12101, Step=12100, Step=121006, …
      Response={AuthenticationResult=NotPerformed; MajorVersion=Default; … }
      IdentityGroup=IdentityGroup:All Groups:WLAN Users:Myco
      ConfigChangeData='Enable Password'='********'\\\,'Password'='********'\\\
      AVPair=task_id=4338, AVPair=timezone=GMT, AVPair=start_time=1319185843
```

The AVPair value is particularly important as it holds incoming TACACS+ authentication, authorisation and accounting attributes. Hence the parser supports their inclusion directly by naming the actual TACACS+ av-pair name, eg “task_id” without having to include the “AVPair=” string, as in the XML snippet below:

```
      <Column Attribute="task_id"/>
```

Example CSV

The example transform given above would result in a CSV with the following columns:

```
      Date,Time,AAA Server,Reason,Acct-Flags,User-Name,NAS-IP-Address,NAS-Port,…
```

Running the Parser

The parser can be used in one of two modes:

- Run. ACS logs are processed, parsed and CSV files generated
- Parse. ACS logs are scanned and a summary of events and attributes is displayed

The parse mode is particularly useful in determining what data is contained within an ACS log, for example after a release from Cisco. It can be used to confirm the presence of new events and/or attributes.

The parser is installed with two XML configuration files, one for transforming ACS5 events into ACS4 equivalent CSVs and another for ACS5 specific CSV that includes all of the available data.

Trouble Shooting

The most likely cause of issue are:

- Incorrect setting of install location in the script file bin\SetFolders.bat
- FTP repository not configured correctly via the ACS5 console
- FTP server not configured correctly
- Firewall/routing preventing ACS or this server reaching the FTP server
- Missing SSH .ini file in bin\getacs5logs folder, or incorrect values (eg ACS SSH console username/password etc)
- 1,000,001 other things ☺

As with all scripted solutions it is possible to test each task manually first, for example login into the ACS console and initiate a backup manually using the commands:

```
backup-logs acslogs repository Extraxi
```

If the above commands work via an interactive login to the ACS console the scripted solution should work provided the documented configuration tasks have been completed.

GetAcs5Logs.exe and AcssysLogToCsv.exe will log diagnostic and debug info with configurable levels.