Ciphers How To Guide

Contents

Overview

By default, a Swivel appliance supports a number of 40- and 56-bit SSL encryption ciphers, in addition to 128-bit ciphers.

In order to disable the weak encryption algorithms, ensure that you apply the Tomcat Ciphers patch on all Swivel Servers. Alternatively, you can enforce 128-bit encryption manually, by modifying the Tomcat configuration to specify which ciphers are permissible on each instance of Swivel.

For instructions on how to apply the Tomcat Ciphers patch - please click the How To Guide

Update

Wednesay 30th May 2018 The following default ciphers have been considered weak/medium: arcfour256,arcfour128,aes128-cbc,3des-cbc

You will need to update /etc/ssh/sshd_config to harder the SSH ciphers:

MACs hmac-sha2-256,hmac-sha2-512

Ciphers aes128-ctr,aes192-ctr,aes256-ctr

HostKey /etc/ssh/ssh_host_rsa_key

KexAlgorithms diffie-hellman-group-exchange-sha256

This is based on next article infosec.mozilla.org/guidelines/openssh#Configuration

Thursday 1st October 2015 The weak Diffie-Hellman keys are also affecting Mobile Provisioning and Update Keys (predominantly on Android). Aswell as affecting Swivel Admin Console Access on Google Chrome and Firefox.

Tuesday 29th September 2015 A patch file that removes all weak ciphers, including the Diffie-Hellman keys has been publically released.

Only the patch can be applied on 2.0.x and 2.1 Appliances.

Please download the Tomcat Ciphers patch here and follow the instructions on how to apply the patch.

Friday 25th September 2015 CVE-2015-2808 reports that RC4 cyphers are now deprecated. The list of supported ciphers has been updated to reflect this.

The list of useable ciphers below has been updated to remove those vulnerable to the logjam vulnerability.

Friday 4th September 2015 There is a known issue, whereby updating Google Chrome (v45) and Mozilla Firefox (v39) no longer supports a weak ephemeral Diffie-Hellman public key. To resolve, the matter, you will need to update the server.xml by following the below section.

The weak Ciphers are:

- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_DSS_WITH_AES_128_CBC_SHA
- SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA
- SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA
- SSL_RSA_WITH_RC4_128_MD5
- SSL_RSA_WITH_RC4_128_SHA

Prerequisites

Swivel appliance version 2.0.16 or earlier running Webmin

Ensure that there are valid backups of the file.

Enforce 128-bit Encryption on Apache Tomcat

You can edit the Tomcat configuration using Webmin. Log in as usual, then go to the Servers -> PINsafe page. Click on "Edit Tomcat Config File".

It is also possible to edit the file using WinSCP or through the CMI. On a Swivel appliance, the file is located at: /usr/local/tomcat/conf/server.xml

There are two places to edit the ciphers within this file on Swivel appliances. Locate the line looking something like the following:

```
<Connector address="0.0.0.0" port="8080" ...
```

Find the end of this line, indicated by the character sequence /**>**. Delete these 2 characters, and insert a new line containing the following:

```
ciphers="TLS_RSA_WITH_AES_128_CBC_SHA, SSL_RSA_WITH_3DES_EDE_CBC_SHA" />
```

Note that this **MUST** be all on one line.

The entire Connector definition should now look something like the following:

```
<Connector address="0.0.0.0" port="8080" scheme="https" secure="true" clientAuth="false" sslProtocol="TLS" keystoreFile="/home/swivel/.keysto
        ciphers="TLS_RSA_WITH_AES_128_CBC_SHA, SSL_RSA_WITH_3DES_EDE_CBC_SHA" />
```

You also need to make the same changes to the Connector with port="8443" (but not the one with port="8181").

Finally, Save the changes and restart Tomcat to apply them.

Restrict Webmin SSL Ciphers

Run the following commands from the Command line accessed through the CMI to make the necessary changes:

```
[admin@primary ~]# echo 'ssl_cipher_list=ALL:!ADH:!LOW:!MEDIUM:!SSLv2:!EXP:+HIGH' >> /etc/webmin/miniserv.conf
[admin@primary ~]# service webmin restart

Stopping Webmin server in /usr/libexec/webmin
Starting Webmin server in /usr/libexec/webmin
```

To test, you should receive the following "alert handshake failure" when you run this command:

```
[admin@primary ~]# openssl s_client -connect localhost.localdomain:10000 -cipher LOW

CONNECTED(00000003)
18339:error:14077410:SSL routines:SSL23_GET_SERVER_HELLO:sslv3 alert handshake failure:s23_clnt.c:470:
```

FAQ

Why is 128-bit encryption used?

AES only supports 128-bit, 192-bit and 256-bit key sizes. Cracking encryption involves cracking the key, and the attacks we see involve finding a weakness in the cipher to drastically reduce the possible space of keys (for a brute force attack). No attack has yet been found for AES-128, so breaking this key requires a search through 2^128, or about 34 undecillion keys. This is a very large number, and AES-128 is secure enough for practical purposes. AES-256 is used by the US military for top secret information.