FREAK Vulnerability - Q&A

What is the FREAK vulnerability?

An issue in the TLS state machine whereby a client system accepts an RSA key with a shorter key length than the originally negotiated key length. (https://technet.microsoft.com/en-us/library/security/ms15-031.aspx).

How is this exploited?

In a man-in-the-middle (MiTM) attack, an attacker could downgrade the key length of an RSA key to EXPORT-grade length in an encrypted TLS session. The attacker could then intercept and decrypt this traffic.

What happens if you explicitly accept low-grade export keys in the cipher suit?

You are vulnerable.

Is ShareScan vulnerable?

It is vulnerable if the web client component has been installed.

How can this be fixed?

You need to make edits to the Tomcat configuration file named `server.xml` as follows:

```
<!-- Define a SSL HTTP/1.1 Connector on port 8443
     This connector uses the JSSE configuration, when using APR, the
     connector should be using the OpenSSL style configuration
     described in the APR documentation -->
<Connector port="443"
     protocol="HTTP/1.1"
     SSLEnabled="true"
     maxThreads="150"
     scheme="https"
     secure="true"
     clientAuth="false"
     sslProtocol="TLS"
     maxHttpHeaderSize="8192"
     minSpareThreads="25"
     enableLookups="false"
     disableUploadTimeout="true"
     acceptCount="100"
```

```
keystoreFile="${catalina.base}/conf/eCopy.key"
URIEncoding="UTF-8"
ciphers="SSL_RSA_WITH_RC4_128_MD5, SSL_RSA_WITH_RC4_128_SHA, TLS_
RSA_WITH_AES_128_CBC_SHA, SSL_RSA_WITH_3DES_EDE_CBC_SHA, SSL_RSA_
WITH_DES_CBC_SHA, SSL_RSA_EXPORT_WITH_RC4_40_MD5, SSL_RSA_EXPORT_
WITH_DES40_CBC_SHA, TLS_EMPTY_RENEGOTIATION_INFO_SCSV"
sslEnabledProtocols="TLSv1,SSLv3,SSLv2Hello"
allowUnsafeLegacyRenegotiation="true"
/>
```

The highlighted line has to be edited and all the RSA_EXPORT have to be removed, so that

```
ciphers="SSL_RSA_WITH_RC4_128_MD5, SSL_RSA_WITH_RC4_128_SHA, TLS_RSA_WITH_
AES_128_CBC_SHA, SSL_RSA_WITH_3DES_EDE_CBC_SHA, SSL_RSA_WITH_DES_CBC_SHA,
SSL_RSA_EXPORT_WITH_RC4_40_MD5, SSL_RSA_EXPORT_WITH_DES40_CBC_SHA, TLS_
EMPTY_RENEGOTIATION_INFO_SCSV"
```

becomes

```
ciphers="SSL_RSA_WITH_RC4_128_MD5, SSL_RSA_WITH_RC4_128_SHA, TLS_RSA_WITH_
AES_128_CBC_SHA, SSL_RSA_WITH_3DES_EDE_CBC_SHA, SSL_RSA_WITH_DES_CBC_SHA,
TLS_EMPTY_RENEGOTIATION_INFO_SCSV".
```

Is there anything to double-check in ShareScan server component as well?

When the PC hosting the ShareScan server acts as a client, ensure that the proper Windows security patches are deployed. For details, see https://technet.microsoft.com/en-us/library/security/ms15-031.aspx.