Redirect on Weak Encryption

Use Case:

SSL protocol layer came under several attacks in last couple years and everything we knew about having strong security has changed. The cipher strength, protocol version and key length can impact the overall security provided by this layer. Hence every customer needs to ensure that they do not accept requests from weak SSL clients compromising overall security posture.

F5 iRules:

```
# iRule Source for less than 128 bits


when HTTP_REQUEST {


   # Check for less than 128 bits of encryption
   if { [SSL::cipher bits] < 128 }{


      # When browser cannot do at least 128 bits of encryption
      #   redirect to a un-encrypted page with an informational error.
      # Set cache control headers to prevent proxies from caching the r
esponse.
      # The cache control headers shouldn't be necessary for a 302,
      #   but it doesn't do any harm setting them.
      HTTP::respond 302 Location "http://10.10.10.10/error/sslerr.html"
Cache-Control No-Cache Pragma No-Cache Connection Close
   }
}


# iRule Source for less than TLS1.1


when HTTP_REQUEST {


       # Check for less than TLSv1.1. This prevents SSLv2, SSLv3, TLSv1
(TLSv1.0 is returned as TLSv1 by [SSL::cipher version]).
       switch -glob [SSL::cipher version] {
              "TLSv1.*" {
```

```
                        # Do nothing and allow the request
                }
                default {
                        # When browser cannot negotiate at least TLSv1.1
                        #       redirect to a unencrypted page with an inf
ormational error.
                        # Set cache control headers to prevent proxies fr
om caching the response.
                        # The cache control headers shouldn't be necessar
y for a 302,
                        #       but it doesn't do any harm setting them.
                        HTTP::respond 302 Location "http://10.10.10.10/er
ror/sslerr.html" Cache-Control No-Cache Pragma No-Cache Connection Clos
e


                        # Log details of the SSL handshake and browser us
er-agent
                        # Consider using High Speed Logging instead to im
prove performance: https://devcentral.f5.com/wiki/iRules.hsl.ashx
                        log local0. "[IP::client_addr]:[TCP::client_port]
:\
                           \[SSL::cipher version\]: [SSL::cipher version]
,\
                           \[SSL::cipher name\]: [SSL::cipher name],\
                           \[SSL::cipher bits\]: [SSL::cipher bits],\
                           U-A: [HTTP::header User-Agent]"
                }
        }
}
```

NetScaler Solution:

```
#Check cipher bits else redirect
add responder action redirect_http redirect '"http://10.10.10.10/error/sslerr.html"'
add responder policy pol_redirect_http CLIENT.SSL.CIPHER_BITS.LT(128) redirect_http

#Check SSL version else redirect
add responder policy pol_check_ssl_version 'CLIENT.SSL.VERSION.EQ(0x301).NOT &&
CLIENT.SSL.VERSION.EQ(0x300).NOT && CLIENT.SSL.VERSION.EQ(0x002).NOT' redirect_http
```

Here the 2 set of actions and policies take care of 2 different use cases for checking the cipher strength and SSL protocol version.