SSL 3.0 Client Tracker

Use Case:

With growing number of vulnerabilities around SSL protocol, SSLv3 is almost rendered useless. In all production deployments, it is advised to disable SSLv3 protocol and before you do that as an administrator, you want to understand what number of clients are actually connecting with SSLv3 as a protocol. This example helps you understand the details of clients connecting to SSL vserver with v3 protocol.

F5 iRules:

```
# iRule to maintain SSL3-only client information in memory

# For performance purposes, it only "records" client information based on clie
nt IP address as a "key" into table

# Rule also will not update information if request is from a client IP address
s that has used SSL3 within the timeout period

# Chad Jenison c.jenison at f5.com


when RULE_INIT {

    #set this value to value in seconds you want to keep ssl3clients in memory
; default is 3600 (1 hour)

    set static::ttl 3600

    set static::honorXffIfExists 1

    set static::xffHeaderName "X-Forwarded-For"

}


when HTTP_REQUEST {

  if {[SSL::cipher version] eq "SSLv3"}{

    if {$static::honorXffIfExists && [HTTP::header exists $static::xffHeaderNa
me]} {

        set requestorip [HTTP::header value $static::xffHeaderName]

        log local0. "SSL3 connection from Proxy: [IP::client_addr] on behalf o
f [HTTP::header value $static::xffHeaderName] **Notify Proxy Admin"

    } else {

        set requestorip [IP::client_addr]

    }
```

```
    if {[table incr -subtable ssl3sourceIPs $requestorip] eq 1}{
        table timeout -subtable ssl3sourceIPs $requestorip $static::ttl
        table set "ssl3host$requestorip" [HTTP::header "Host"] $static::ttl
        table set "ssl3useragent$requestorip" [HTTP::header "User-Agent"] $sta
tic::ttl
        table set "sslcipher$requestorip" [SSL::cipher name] $static::ttl
        log local0. "SSL Cipher Used: [SSL::cipher name]"
    } else {
        table timeout -subtable ssl3sourceIPs $requestorip $static::ttl
        table timeout "ssl3host$requestorip" $static::ttl
        table timeout "ssl3useragent$requestorip" $static::ttl
        table timeout "sslcipher$requestorip" $static::ttl
    }
  }
  if {[HTTP::uri] starts_with "/ssl3lookup/"}{
     set ssl3clienttable "<table border=\"1\"><tr><th>Source IP</th><th>Host H
eader</th><th>User-Agent</th><th>Geolocation</th><th>SSL Cipher Used</th><th>H
TTP Requests</th></tr>"
     foreach clientip [table keys -subtable ssl3sourceIPs] {
        append ssl3clienttable "<tr><td>$clientip</td><td>[table lookup "ssl3h
ost$clientip"]</td><td>[table lookup "ssl3useragent$clientip"]</td><td>[table
lookup "sslcipher$clientip"]</td><td>[whereis $clientip continent]:[whereis $c
lientip country]:[whereis $clientip state]</td><td>[table lookup -subtable ssl
3sourceIPs $clientip]</td></tr>"
     }
     append ssl3clienttable "</table>"
     HTTP::respond 200 content "<HTML><HEAD><TITLE>SSL3 Client Table</TITLE></
HEAD><BODY>$ssl3clienttable</BODY></HTML>"
     log local0. "Got Magic Request"
  }
}
```

URL: https://devcentral.f5.com/codeshare/ssl-30-client-tracker

NetScaler Solution:

```
set syslogParams -userDefinedAuditlog YES
```

```
add auditmessageaction log_ssl_v3_users INFORMATIONAL '"[SSLv3]
Client IP : "+CLIENT.IP.SRC+ " Accessing the URL: "+HTTP.REQ.URL' -
bypassSafetyCheck YES
```

```
add responder policy log_ssl_v3_pol CLIENT.SSL.VERSION.EQ(0x300)
NOOP -logAction log_ssl_v3_users
```

NetScaler provides you a simple infrastructure to log the traffic details using “auditmessageaction” infrastructure. You can add as many headers and other L2 to L7 details you want to log about the incoming request here by adding respective expressions to the action. Bind the action to Responder policy which can then be bound to VSERVER (where you want to enable explicit logging) or to Global responder bind point.