ПЕРЕНОС НАБОРОВ АЛГОРИТМОВ ШИФРОВАНИЯ ИЗ WIRESHARK В BIG-IP

Джон Вагнон (John Wagnon), старший разработчик решений F5 Networks, США

Многие из нас пользуются Wireshark для перехвата и анализа сетевого трафика. При работе с SSL/TLS-трафиком Wireshark прекрасно справляется с отображением наборов алгоритмов шифрования, передаваемых клиентом и выбираемых сервером в ходе конкретного сеанса SSL/TLS. Поскольку названия этих наборов не стандартизированы, в разных системах они отображаются немного по-разному. Например, BIG-IP показывает их следующим образом:

DHE-RSA-AES256-GCM-SHA384 256 TLS1.2 Native AES-GCM SHA384 EDH/RSA

Различные браузеры также по-своему отображают наборы, используемые для определенных веб-страниц. Например, Google Chrome покажет такое сообщение:

Secure connection

The connection to this site is encrypted and authenticated using a strong protocol (TLS 1.2), a strong key exchange (ECDHE_RSA with P-256), and a strong cipher (AES_128_GCM).

А вот пример сообщения в Firefox:

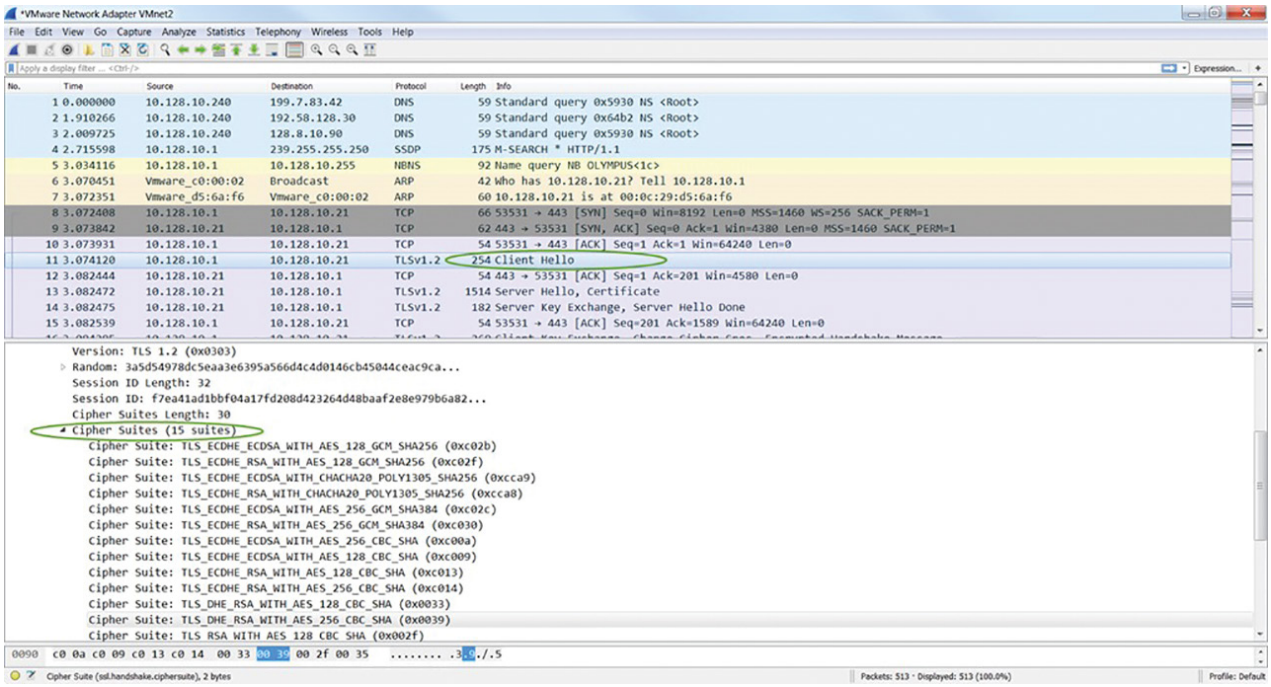
Connection Encrypted (TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256, 128 bit keys, TLS 1.2)

The page you are viewing was encrypted before being transmitted over the Internet.

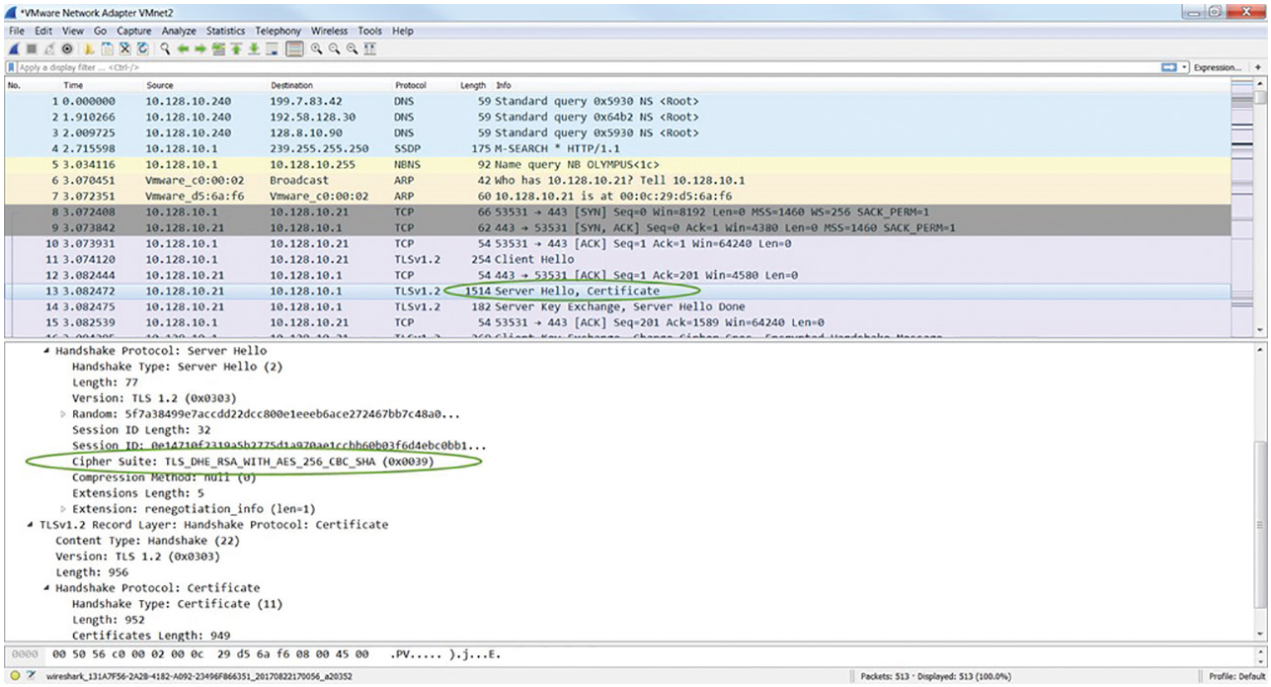
Encryption makes it difficult for unauthorized people to view information traveling between computers. It is therefore unlikely that anyone read this page as it traveled across the network.

Как видите, иногда бывает сложно однозначно установить набор алгоритмов шифрования, используемый для определенной страницы в ходе сеанса. С этой задачей прекрасно справится Wireshark. Его можно использовать для перехвата фактических SSL/TLS-транзакций между клиентом и сервером с формированием списка наборов алгоритмов шифрования, предлагаемых клиентом, а также указанием набора, выбранного сервером. Каждому из них присваивается некое шестнадцатеричное значение. Его очень удобно использовать при отслеживании каждого набора.

У меня установлен BIG-IP v12.0. При доступе к тестовому веб-приложению я перехватил при помощи Wireshark установку связи между клиентом и сервером (сообщения Client Hello и Server Hello). В моем BIG-IP был включен стандартный набор алгоритмов шифрования SSL-профиля. Вот содержание перехваченного Wireshark запроса на установление связи от клиента:



И хотя на данном скриншоте этого не видно, браузер отправил на сервер 15 различных наборов алгоритмов шифрования на выбор. Обратите внимание на шестнадцатеричное значение, указанное рядом с каждым набором. Особо стоит обратить внимание на набор алгоритмов со значением 0x0039. Именно его в конечном итоге выберет BIG-IP для данного сеанса. Вот содержание ответа сервера на запрос клиента на установление связи, перехваченного Wireshark (вскоре после перехвата запроса от клиента):



Здесь видно, какой набор алгоритмов шифрования выбрал BIG-IP (сервер) для данного сеанса. Этот набор алгоритмов тоже имеет шестнадцатеричное значение – 0x0039. Как я уже говорил, перехват трафика при помощи Wireshark позволяет узнавать, что входит в набор алгоритмов шифрования (TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x0039)), но это не всегда однозначно соответствует наименованиям, используемым в списке набора алгоритмов BIG-IP. Вот список стандратных наборов алгоритмов шифрования, используемых в BIG-IP (v12.0), полученный при помощи следующей команды:

```
tmm --clientciphers DEFAULT
```

```
[root@wagnon:Active:Standalone] config # tmm --clientciphers DEFAULT
      ID  SUITE                          BITS PROT    METHOD  CIPHER    MAC     KEYX
 0:  159  DHE-RSA-AES256-GCM-SHA384       256  TLS1.2  Native  AES-GCM  SHA384  EDH/RSA
 1:  158  DHE-RSA-AES128-GCM-SHA256       128  TLS1.2  Native  AES-GCM  SHA256  EDH/RSA
 2:  107  DHE-RSA-AES256-SHA256           256  TLS1.2  Native  AES      SHA256  EDH/RSA
 3:   57  DHE-RSA-AES256-SHA              256  TLS1    Native  AES      SHA     EDH/RSA
 4:   57  DHE-RSA-AES256-SHA              256  TLS1.1  Native  AES      SHA     EDH/RSA
 5:   57  DHE-RSA-AES256-SHA              256  TLS1.2  Native  AES      SHA     EDH/RSA
 6:   57  DHE-RSA-AES256-SHA              256  DTLS1   Native  AES      SHA     EDH/RSA
 7:  103  DHE-RSA-AES128-SHA256           128  TLS1.2  Native  AES      SHA256  EDH/RSA
 8:   51  DHE-RSA-AES128-SHA              128  TLS1    Native  AES      SHA     EDH/RSA
 9:   51  DHE-RSA-AES128-SHA              128  TLS1.1  Native  AES      SHA     EDH/RSA
10:   51  DHE-RSA-AES128-SHA              128  TLS1.2  Native  AES      SHA     EDH/RSA
11:   51  DHE-RSA-AES128-SHA              128  DTLS1   Native  AES      SHA     EDH/RSA
12:   22  DHE-RSA-DES-CBC3-SHA            168  TLS1    Native  DES      SHA     EDH/RSA
13:   22  DHE-RSA-DES-CBC3-SHA            168  TLS1.1  Native  DES      SHA     EDH/RSA
14:   22  DHE-RSA-DES-CBC3-SHA            168  TLS1.2  Native  DES      SHA     EDH/RSA
15:   22  DHE-RSA-DES-CBC3-SHA            168  DTLS1   Native  DES      SHA     EDH/RSA
16:  157  AES256-GCM-SHA384               256  TLS1.2  Native  AES-GCM  SHA384  RSA
17:  156  AES128-GCM-SHA256               128  TLS1.2  Native  AES-GCM  SHA256  RSA
18:   61  AES256-SHA256                   256  TLS1.2  Native  AES      SHA256  RSA
19:   53  AES256-SHA                      256  TLS1    Native  AES      SHA     RSA
20:   53  AES256-SHA                      256  TLS1.1  Native  AES      SHA     RSA
21:   53  AES256-SHA                      256  TLS1.2  Native  AES      SHA     RSA
22:   53  AES256-SHA                      256  DTLS1   Native  AES      SHA     RSA
23:   60  AES128-SHA256                   128  TLS1.2  Native  AES      SHA256  RSA
24:   47  AES128-SHA                      128  TLS1    Native  AES      SHA     RSA
25:   47  AES128-SHA                      128  TLS1.1  Native  AES      SHA     RSA
26:   47  AES128-SHA                      128  TLS1.2  Native  AES      SHA     RSA
27:   47  AES128-SHA                      128  DTLS1   Native  AES      SHA     RSA
28:   10  DES-CBC3-SHA                    168  TLS1    Native  DES      SHA     RSA
29:   10  DES-CBC3-SHA                    168  TLS1.1  Native  DES      SHA     RSA
30:   10  DES-CBC3-SHA                    168  TLS1.2  Native  DES      SHA     RSA
31:   10  DES-CBC3-SHA                    168  DTLS1   Native  DES      SHA     RSA
32: 49200 ECDHE-RSA-AES256-GCM-SHA384      256  TLS1.2  Native  AES-GCM  SHA384  ECDHE_RSA
33: 49199 ECDHE-RSA-AES128-GCM-SHA256      128  TLS1.2  Native  AES-GCM  SHA256  ECDHE_RSA
34: 49192 ECDHE-RSA-AES256-SHA384          256  TLS1.2  Native  AES      SHA384  ECDHE_RSA
35: 49172 ECDHE-RSA-AES256-CBC-SHA         256  TLS1    Native  AES      SHA     ECDHE_RSA
36: 49172 ECDHE-RSA-AES256-CBC-SHA         256  TLS1.1  Native  AES      SHA     ECDHE_RSA
37: 49172 ECDHE-RSA-AES256-CBC-SHA         256  TLS1.2  Native  AES      SHA     ECDHE_RSA
38: 49191 ECDHE-RSA-AES128-SHA256          128  TLS1.2  Native  AES      SHA256  ECDHE_RSA
39: 49171 ECDHE-RSA-AES128-CBC-SHA         128  TLS1    Native  AES      SHA     ECDHE_RSA
40: 49171 ECDHE-RSA-AES128-CBC-SHA         128  TLS1.1  Native  AES      SHA     ECDHE_RSA
41: 49171 ECDHE-RSA-AES128-CBC-SHA         128  TLS1.2  Native  AES      SHA     ECDHE_RSA
42: 49170 ECDHE-RSA-DES-CBC3-SHA           168  TLS1    Native  DES      SHA     ECDHE_RSA
43: 49170 ECDHE-RSA-DES-CBC3-SHA           168  TLS1.1  Native  DES      SHA     ECDHE_RSA
44: 49170 ECDHE-RSA-DES-CBC3-SHA           168  TLS1.2  Native  DES      SHA     ECDHE_RSA
```

К сожалению, BIG-IP не формирует списков шестнадцатеричных значений наборов шифров по умолчанию (с использованием вышеуказанной команды). Тем не менее, для получения списка наборов с их шестнадцатеричными значениями можно воспользоваться следующей командой (при этом буква V обязательно должна быть в верхнем регистре):

```
openssl ciphers –V DEFAULT
```

Вот скриншот с тем же списком алгоритмов шифрования, полученный при помощи команды openssl:

```
[root@wagnon:Active:Standalone] config # openssl ciphers -V DEFAULT
        0xC0,0x30 - ECDHE-RSA-AES256-GCM-SHA384 TLSv1.2 Kx=ECDH     Au=RSA  Enc=AESGCM(256) Mac=AEAD
        0xC0,0x2C - ECDHE-ECDSA-AES256-GCM-SHA384 TLSv1.2 Kx=ECDH     Au=ECDSA Enc=AESGCM(256) Mac=AEAD
        0xC0,0x28 - ECDHE-RSA-AES256-SHA384 TLSv1.2 Kx=ECDH     Au=RSA  Enc=AES(256)  Mac=SHA384
        0xC0,0x24 - ECDHE-ECDSA-AES256-SHA384 TLSv1.2 Kx=ECDH     Au=ECDSA Enc=AES(256)  Mac=SHA384
        0xC0,0x14 - ECDHE-RSA-AES256-SHA     SSLv3 Kx=ECDH     Au=RSA  Enc=AES(256)  Mac=SHA1
        0xC0,0x0A - ECDHE-ECDSA-AES256-SHA   SSLv3 Kx=ECDH     Au=ECDSA Enc=AES(256)  Mac=SHA1
        0x00,0xA3 - DHE-DSS-AES256-GCM-SHA384 TLSv1.2 Kx=DH       Au=DSS  Enc=AESGCM(256) Mac=AEAD
        0x00,0x9F - DHE-RSA-AES256-GCM-SHA384 TLSv1.2 Kx=DH       Au=RSA  Enc=AESGCM(256) Mac=AEAD
        0x00,0x6B - DHE-RSA-AES256-SHA256    TLSv1.2 Kx=DH       Au=RSA  Enc=AES(256)  Mac=SHA256
        0x00,0x6A - DHE-DSS-AES256-SHA256    TLSv1.2 Kx=DH       Au=DSS  Enc=AES(256)  Mac=SHA256
        0x00,0x39 - DHE-RSA-AES256-SHA       SSLv3 Kx=DH       Au=RSA  Enc=AES(256)  Mac=SHA1
        0x00,0x38 - DHE-DSS-AES256-SHA       SSLv3 Kx=DH       Au=DSS  Enc=AES(256)  Mac=SHA1
        0x00,0x88 - DHE-RSA-CAMELLIA256-SHA SSLv3 Kx=DH       Au=RSA  Enc=Camellia(256) Mac=SHA1
        0x00,0x87 - DHE-DSS-CAMELLIA256-SHA SSLv3 Kx=DH       Au=DSS  Enc=Camellia(256) Mac=SHA1
        0xC0,0x32 - ECDH-RSA-AES256-GCM-SHA384 TLSv1.2 Kx=ECDH/RSA Au=ECDH Enc=AESGCM(256) Mac=AEAD
        0xC0,0x2E - ECDH-ECDSA-AES256-GCM-SHA384 TLSv1.2 Kx=ECDH/ECDSA Au=ECDH Enc=AESGCM(256) Mac=AEAD
        0xC0,0x2A - ECDH-RSA-AES256-SHA384   TLSv1.2 Kx=ECDH/RSA Au=ECDH Enc=AES(256)  Mac=SHA384
        0xC0,0x26 - ECDH-ECDSA-AES256-SHA384 TLSv1.2 Kx=ECDH/ECDSA Au=ECDH Enc=AES(256)  Mac=SHA384
        0xC0,0x0F - ECDH-RSA-AES256-SHA      SSLv3 Kx=ECDH/RSA Au=ECDH Enc=AES(256)  Mac=SHA1
        0xC0,0x05 - ECDH-ECDSA-AES256-SHA    SSLv3 Kx=ECDH/ECDSA Au=ECDH Enc=AES(256)  Mac=SHA1
        0x00,0x9D - AES256-GCM-SHA384        TLSv1.2 Kx=RSA      Au=RSA  Enc=AESGCM(256) Mac=AEAD
        0x00,0x3D - AES256-SHA256            TLSv1.2 Kx=RSA      Au=RSA  Enc=AES(256)  Mac=SHA256
        0x00,0x35 - AES256-SHA               SSLv3 Kx=RSA      Au=RSA  Enc=AES(256)  Mac=SHA1
        0x00,0x84 - CAMELLIA256-SHA          SSLv3 Kx=RSA      Au=RSA  Enc=Camellia(256) Mac=SHA1
        0x00,0x8D - PSK-AES256-CBC-SHA       SSLv3 Kx=PSK      Au=PSK  Enc=AES(256)  Mac=SHA1
        0xC0,0x2F - ECDHE-RSA-AES128-GCM-SHA256 TLSv1.2 Kx=ECDH     Au=RSA  Enc=AESGCM(128) Mac=AEAD
        0xC0,0x2B - ECDHE-ECDSA-AES128-GCM-SHA256 TLSv1.2 Kx=ECDH     Au=ECDSA Enc=AESGCM(128) Mac=AEAD
        0xC0,0x27 - ECDHE-RSA-AES128-SHA256 TLSv1.2 Kx=ECDH     Au=RSA  Enc=AES(128)  Mac=SHA256
        0xC0,0x23 - ECDHE-ECDSA-AES128-SHA256 TLSv1.2 Kx=ECDH     Au=ECDSA Enc=AES(128)  Mac=SHA256
        0xC0,0x13 - ECDHE-RSA-AES128-SHA     SSLv3 Kx=ECDH     Au=RSA  Enc=AES(128)  Mac=SHA1
        0xC0,0x09 - ECDHE-ECDSA-AES128-SHA   SSLv3 Kx=ECDH     Au=ECDSA Enc=AES(128)  Mac=SHA1
        0x00,0xA2 - DHE-DSS-AES128-GCM-SHA256 TLSv1.2 Kx=DH       Au=DSS  Enc=AESGCM(128) Mac=AEAD
        0x00,0x9E - DHE-RSA-AES128-GCM-SHA256 TLSv1.2 Kx=DH       Au=RSA  Enc=AESGCM(128) Mac=AEAD
        0x00,0x67 - DHE-RSA-AES128-SHA256    TLSv1.2 Kx=DH       Au=RSA  Enc=AES(128)  Mac=SHA256
        0x00,0x40 - DHE-DSS-AES128-SHA256    TLSv1.2 Kx=DH       Au=DSS  Enc=AES(128)  Mac=SHA256
        0x00,0x33 - DHE-RSA-AES128-SHA       SSLv3 Kx=DH       Au=RSA  Enc=AES(128)  Mac=SHA1
        0x00,0x32 - DHE-DSS-AES128-SHA       SSLv3 Kx=DH       Au=DSS  Enc=AES(128)  Mac=SHA1
        0x00,0x9A - DHE-RSA-SEED-SHA         SSLv3 Kx=DH       Au=RSA  Enc=SEED(128) Mac=SHA1
        0x00,0x99 - DHE-DSS-SEED-SHA         SSLv3 Kx=DH       Au=DSS  Enc=SEED(128) Mac=SHA1
        0x00,0x45 - DHE-RSA-CAMELLIA128-SHA SSLv3 Kx=DH       Au=RSA  Enc=Camellia(128) Mac=SHA1
        0x00,0x44 - DHE-DSS-CAMELLIA128-SHA SSLv3 Kx=DH       Au=DSS  Enc=Camellia(128) Mac=SHA1
        0xC0,0x31 - ECDH-RSA-AES128-GCM-SHA256 TLSv1.2 Kx=ECDH/RSA Au=ECDH Enc=AESGCM(128) Mac=AEAD
        0xC0,0x2D - ECDH-ECDSA-AES128-GCM-SHA256 TLSv1.2 Kx=ECDH/ECDSA Au=ECDH Enc=AESGCM(128) Mac=AEAD
        0xC0,0x29 - ECDH-RSA-AES128-SHA256   TLSv1.2 Kx=ECDH/RSA Au=ECDH Enc=AES(128)  Mac=SHA256
        0xC0,0x25 - ECDH-ECDSA-AES128-SHA256 TLSv1.2 Kx=ECDH/ECDSA Au=ECDH Enc=AES(128)  Mac=SHA256
        0xC0,0x0E - ECDH-RSA-AES128-SHA      SSLv3 Kx=ECDH/RSA Au=ECDH Enc=AES(128)  Mac=SHA1
        0xC0,0x04 - ECDH-ECDSA-AES128-SHA    SSLv3 Kx=ECDH/ECDSA Au=ECDH Enc=AES(128)  Mac=SHA1
        0x00,0x9C - AES128-GCM-SHA256        TLSv1.2 Kx=RSA      Au=RSA  Enc=AESGCM(128) Mac=AEAD
        0x00,0x3C - AES128-SHA256            TLSv1.2 Kx=RSA      Au=RSA  Enc=AES(128)  Mac=SHA256
        0x00,0x2F - AES128-SHA               SSLv3 Kx=RSA      Au=RSA  Enc=AES(128)  Mac=SHA1
        0x00,0x96 - SEED-SHA                 SSLv3 Kx=RSA      Au=RSA  Enc=SEED(128) Mac=SHA1
        0x00,0x41 - CAMELLIA128-SHA          SSLv3 Kx=RSA      Au=RSA  Enc=Camellia(128) Mac=SHA1
        0x00,0x8C - PSK-AES128-CBC-SHA       SSLv3 Kx=PSK      Au=PSK  Enc=AES(128)  Mac=SHA1
        0xC0,0x11 - ECDHE-RSA-RC4-SHA        SSLv3 Kx=ECDH     Au=RSA  Enc=RC4(128)  Mac=SHA1
        0xC0,0x07 - ECDHE-ECDSA-RC4-SHA      SSLv3 Kx=ECDH     Au=ECDSA Enc=RC4(128)  Mac=SHA1
        0xC0,0x0C - ECDH-RSA-RC4-SHA         SSLv3 Kx=ECDH/RSA Au=ECDH Enc=RC4(128)  Mac=SHA1
        0xC0,0x02 - ECDH-ECDSA-RC4-SHA       SSLv3 Kx=ECDH/ECDSA Au=ECDH Enc=RC4(128)  Mac=SHA1
        0x00,0x05 - RC4-SHA                  SSLv3 Kx=RSA      Au=RSA  Enc=RC4(128)  Mac=SHA1
        0x00,0x04 - RC4-MD5                  SSLv3 Kx=RSA      Au=RSA  Enc=RC4(128)  Mac=MD5
        0x00,0x8A - PSK-RC4-SHA              SSLv3 Kx=PSK      Au=PSK  Enc=RC4(128)  Mac=SHA1
        0xC0,0x12 - ECDHE-RSA-DES-CBC3-SHA   SSLv3 Kx=ECDH     Au=RSA  Enc=3DES(168) Mac=SHA1
        0xC0,0x08 - ECDHE-ECDSA-DES-CBC3-SHA SSLv3 Kx=ECDH     Au=ECDSA Enc=3DES(168) Mac=SHA1
        0x00,0x16 - EDH-RSA-DES-CBC3-SHA     SSLv3 Kx=DH       Au=RSA  Enc=3DES(168) Mac=SHA1
        0x00,0x13 - EDH-DSS-DES-CBC3-SHA     SSLv3 Kx=DH       Au=DSS  Enc=3DES(168) Mac=SHA1
        0xC0,0x0D - ECDH-RSA-DES-CBC3-SHA    SSLv3 Kx=ECDH/RSA Au=ECDH Enc=3DES(168) Mac=SHA1
        0xC0,0x03 - ECDH-ECDSA-DES-CBC3-SHA SSLv3 Kx=ECDH/ECDSA Au=ECDH Enc=3DES(168) Mac=SHA1
        0x00,0x0A - DES-CBC3-SHA             SSLv3 Kx=RSA      Au=RSA  Enc=3DES(168) Mac=SHA1
        0x00,0x8B - PSK-3DES-EDE-CBC-SHA     SSLv3 Kx=PSK      Au=PSK  Enc=3DES(168) Mac=SHA1
        0x00,0x15 - EDH-RSA-DES-CBC-SHA      SSLv3 Kx=DH       Au=RSA  Enc=DES(56)   Mac=SHA1
        0x00,0x12 - EDH-DSS-DES-CBC-SHA      SSLv3 Kx=DH       Au=DSS  Enc=DES(56)   Mac=SHA1
        0x00,0x09 - DES-CBC-SHA              SSLv3 Kx=RSA      Au=RSA  Enc=DES(56)   Mac=SHA1
        0x00,0x14 - EXP-EDH-RSA-DES-CBC-SHA SSLv3 Kx=DH(512)  Au=RSA  Enc=DES(40)   Mac=SHA1 export
        0x00,0x11 - EXP-EDH-DSS-DES-CBC-SHA SSLv3 Kx=DH(512)  Au=DSS  Enc=DES(40)   Mac=SHA1 export
        0x00,0x08 - EXP-DES-CBC-SHA          SSLv3 Kx=RSA(512) Au=RSA  Enc=DES(40)   Mac=SHA1 export
        0x00,0x06 - EXP-RC2-CBC-MD5          SSLv3 Kx=RSA(512) Au=RSA  Enc=RC2(40)   Mac=MD5  export
        0x00,0x03 - EXP-RC4-MD5              SSLv3 Kx=RSA(512) Au=RSA  Enc=RC4(40)   Mac=MD5  export
```

В нем видны соответствующие шестнадцатеричные значения. И, конечно же, этими данными можно воспользоваться для определения шестнадцатеричного значения набора алгоритмов шифрования, перехваченного Wireshark, чтобы узнать, какой именно алгоритм был выбран. Более легкую для восприятия версию алгоритмов шифрования с соответствующими

шестнадцатеричными значениями можно найти по этой ссылке: https://support.f5.com/csp/article/K13156. Там приведен список наборов, используемых BIG-IP, с их шестнадцатеричными значениями.

Вот скриншот с наборами алгоритмов шифрования BIG-IP (v12.0.0 - 12.1.2) с этой страницы. Я специально обвел 0x0039, чтобы дать возможность отследить использование одного и того же набора алгоритмов на протяжении данного сеанса.

BIG-IP 12.0.0 - 12.1.2

In BIG-IP 12.0.0 - 12.1.2, the default Client and Server SSL profiles allow the following SSL ciphers:

Cipher Suite (hex value)	Bits	Protocols	Key Exchange	Authentication	Cipher	MAC
DHE-RSA-AES256-GCM-SHA384 (0x9f)	256	TLS1.2	EDH	RSA	AES-GCM	SHA384
DHE-RSA-AES128-GCM-SHA256 (0x9e)	128	TLS1.2	EDH	RSA	AES-GCM	SHA256
DHE-RSA-AES256-SHA256 (0x6b)	256	TLS1.2	EDH	RSA	AES	SHA256
DHE-RSA-AES256-SHA (0x39)	256	TLS1, TLS1.1, TLS1.2, DTLS1	EDH	RSA	AES	SHA
DHE-RSA-AES128-SHA256 (0x67)	128	TLS1.2	EDH	RSA	AES	SHA256
DHE-RSA-AES128-SHA (0x33)	128	TLS1, TLS1.1, TLS1.2, DTLS1	EDH	RSA	AES	SHA
DHE-RSA-DES-CBC3-SHA (0x16)	168	TLS1, TLS1.1, TLS1.2, DTLS1	EDH	RSA	DES	SHA
AES256-GCM-SHA384 (0x9d)	256	TLS1.2	RSA	RSA	AES-GCM	SHA384
AES128-GCM-SHA256 (0x9c)	128	TLS1.2	RSA	RSA	AES-GCM	SHA256
AES256-SHA256 (0x3d)	256	TLS1.2	RSA	RSA	AES	SHA256
AES256-SHA (0x35)	256	TLS1, TLS1.1, TLS1.2, DTLS1	RSA	RSA	AES	SHA
AES128-SHA256 (0x3c)	128	TLS1.2	RSA	RSA	AES	SHA256
AES128-SHA (0x2f)	128	TLS1, TLS1.1, TLS1.2, DTLS1	RSA	RSA	AES	SHA
DES-CBC3-SHA (0xa)	168	TLS1, TLS1.1, TLS1.2, DTLS1	RSA	RSA	DES	SHA

Группа компаний БАКОТЕК – официальный дистрибьютор F5 Networks в Украине, Азербайджане, Республике Беларусь, Грузии, Армении и Молдове.
https://bakotech.com, f5@bakotech.com, +38 044 273 33 33.

F5 Networks, Inc.
401 Elliott Avenue West, Seattle, WA 98119
888-882-4447 f5.com

Americas
info@f5.com

Asia-Pacific
apacinfo@f5.com

Europe/Middle-East/Africa
emeainfo@f5.com

Japan
f5j-info@f5.com