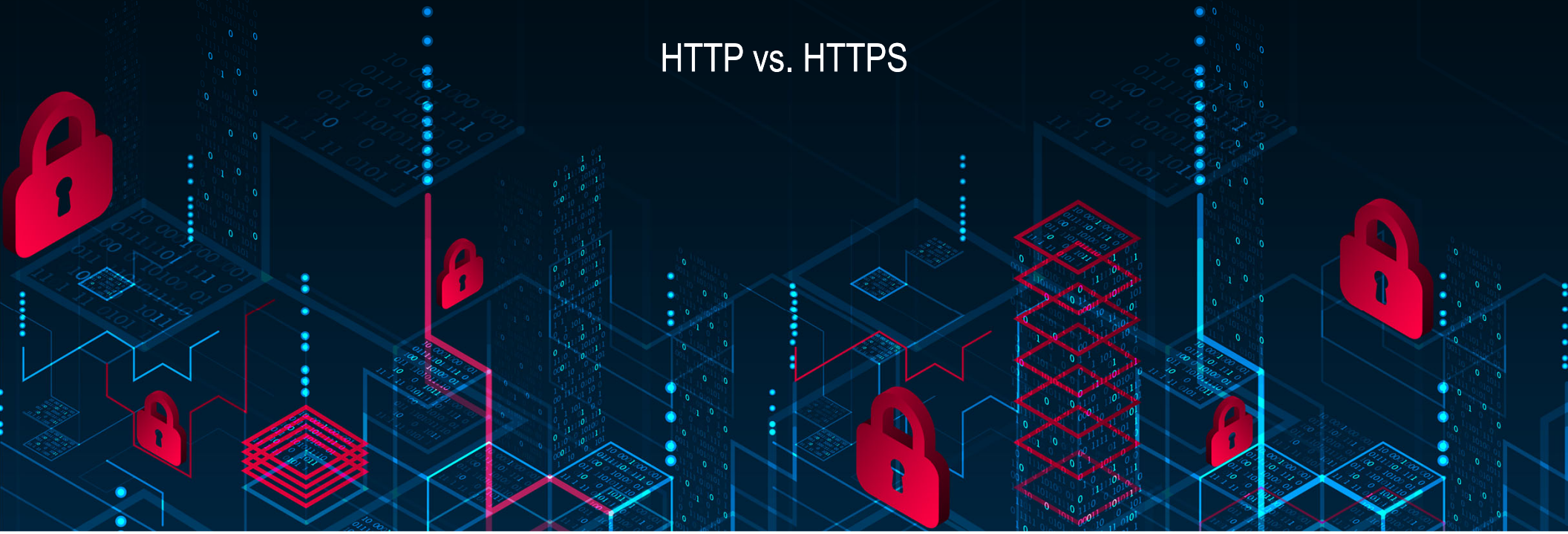
CPR E 431

BASICS OF INFORMATION SYSTEM SECURITY

Internet Security Protocols and Standards

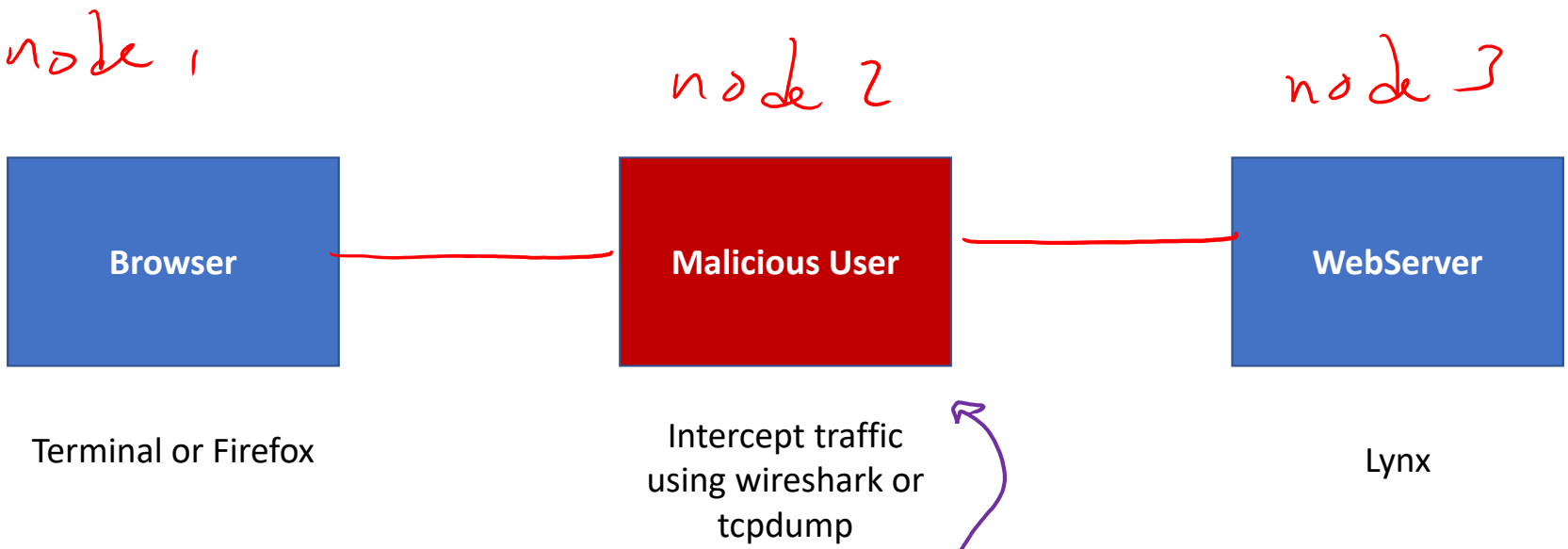
HTTP vs. HTTPS

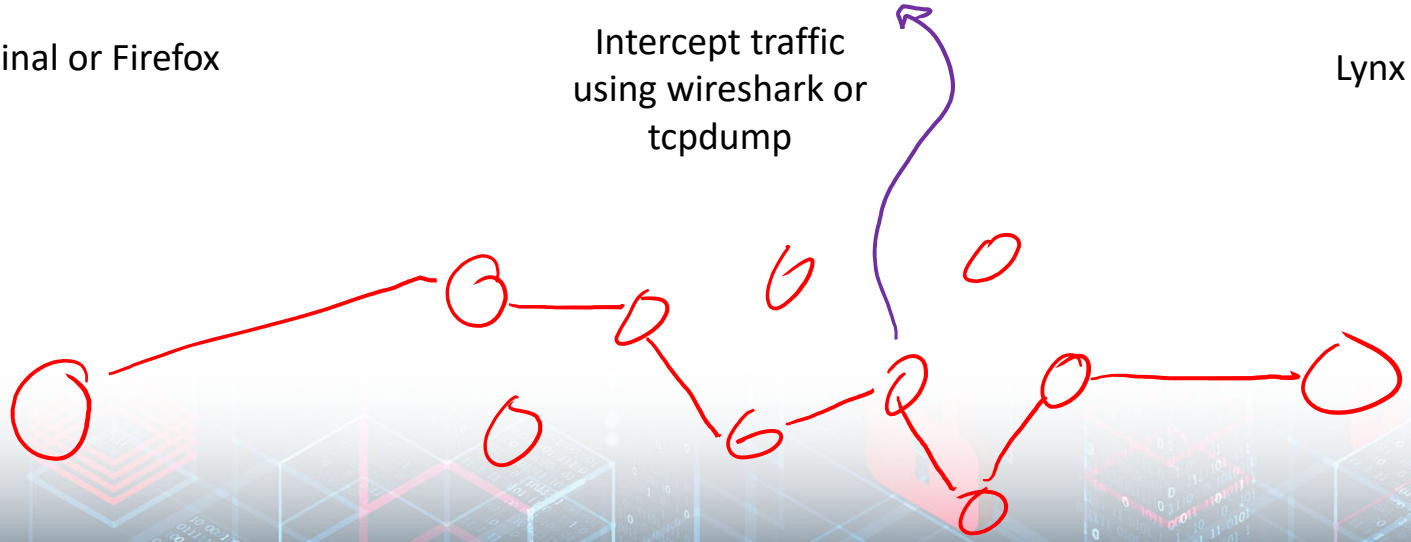


Video summary

- How does HTTP Work?
- HTTP Security Issues
- Using HTTPS
- Introducing SSL/TLS

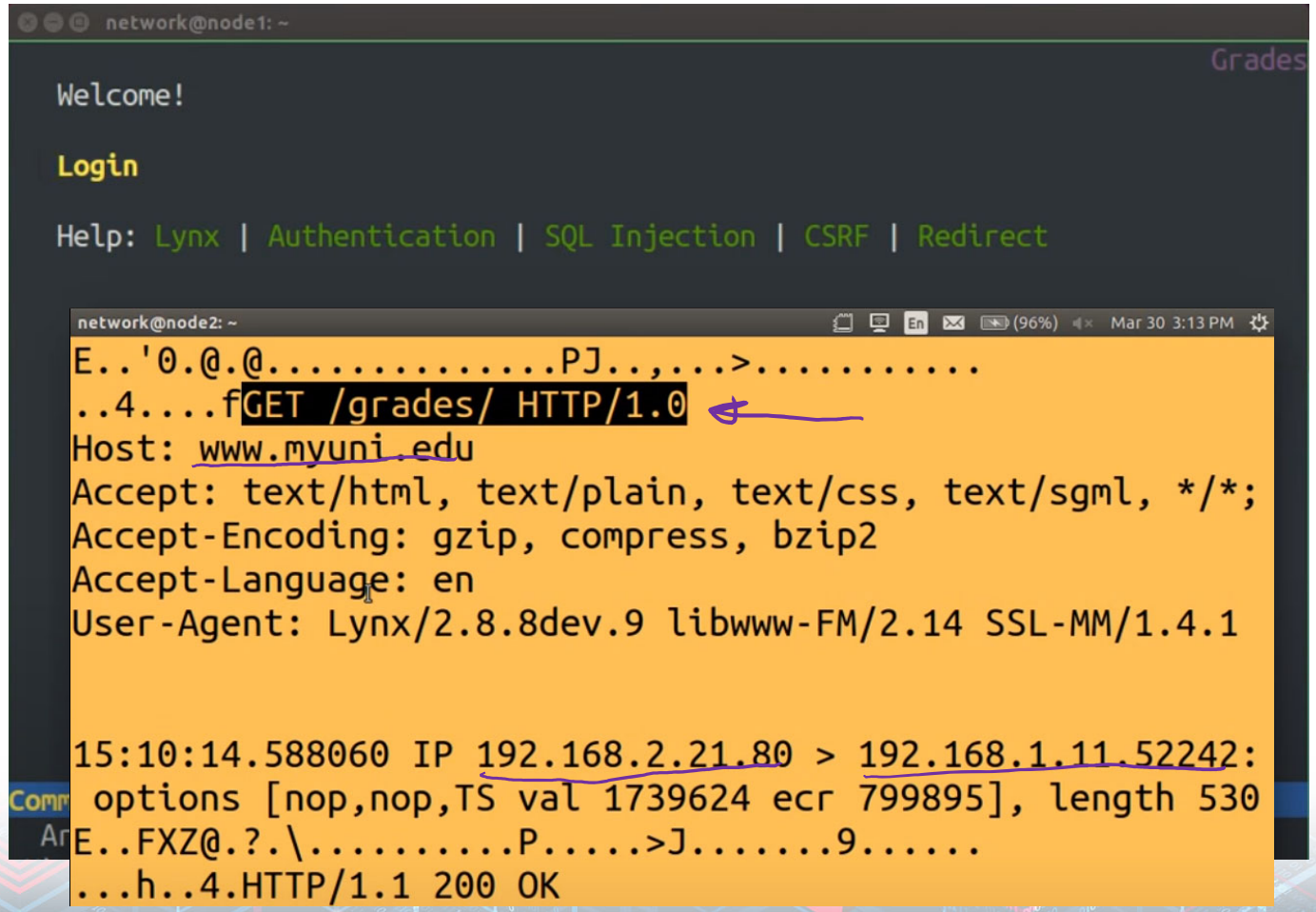
How Does HTTP Work?





How Does HTTP Work?

What malicious user can see on node 2



How Does HTTP Work?

Log in to myuni.edu

```
network@node1: ~
<<<                                                    Grades
LOGIN

Username: 5000000000_____

Password: *******_______

Login

Help: Lynx | Authentication | SQL Injection | CSRF | Redirect
```

```
network@node1: ~
<<<                                                    Grades
Welcome 5000000000

View grades

Logout

Help: Lynx | Authentication | SQL Injection | CSRF | Redirect
```

How Does HTTP Work?

What malicious user can see on node 2

```
E.....@.@...............Pq.....+......@.....
....WaPOST /grades/login.php HTTP/1.0
Host: www.myuni.edu
Accept: text/html, text/plain, text/css, text/sgml, */*;q=0.01
Accept-Encoding: gzip, compress, bzip2
Accept-Language: en
Pragma: no-cache
Cache-Control: no-cache
User-Agent: Lynx/2.8.8dev.9 libwww-FM/2.14 SSL-MM/1.4.1 GNUTLS/2
Referer: http://www.myuni.edu/grades/login.php
```

```
user_name=5000000000&password=student&submit=Login
15:13:46.978004 IP 192.168.2.21.80 > 192.168.1.11.52244: Flags [P.]
, options [nop,nop,TS val 1791843 ecr 853004], length 961
E...|.@.?.6..........P....+.q........E.....
..HTTP/1.1 302 Found
```

What are the HTTP security issues?

- No data confidentiality
- No data user confidentiality
- No mutual authentication

HTTPS

What is the Solution? Use HTTPS

1 ▸ Data transmitted between browser and server is confidential: encryption with HTTPS

2 ▸ Browser sure it is communicating with intended server: digital certificates → Digital Signature

HTTP ▸ Server sure it is communicating with intended user: password authentication, session management

▸ Actions performed by server (engine) are appropriate: authentication, access control

3 ▸ Actions of user (of browser) are kept private: anonymity services

HTTPS

- HTTPS: HTTP over SSL (or TLS)
- URL uses https://
- Web server listens on port 443
- Encrypt: URL of requested document, contents of document, contents of browser forms, cookies, contents of HTTP header
- Server is authenticated using certificate (using SSL)
- Client is authenticated using password (using HTTP)

80 HTTP

SSL / TLS

SSL v3

mutual auth.

SSL and TLS

- Secure Sockets Layer (SSL) originated in Netscape web browser
- Transport Layer Security (TLS) standardised by IETF
- SSLv3 and TLS are almost the same
- SSL provides security services to application layer protocols using TCP
- SSL architecture consists of multiple protocols

How does SSL/TLS Work?

client

User A

PR_A, PU_A, PU_B

Server

User B

PR_B, PU_B, PU_A

K_{CS}

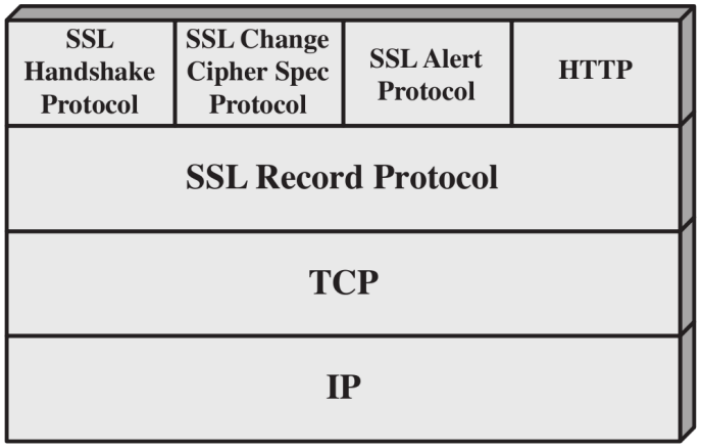
$C = E_{RSA}(PU_A, K_{CS})$

$K_{CS} = D_{RSA}(PR_A, C)$

credit card information CC

$E_{AES}(K_{CS}, CC) \xrightarrow{C} D_{AES}(K_{CS}, C)$

SSL Architecture



Record: provides confidentiality and message integrity

Handshake: authenticate entities, negotiate parameter values

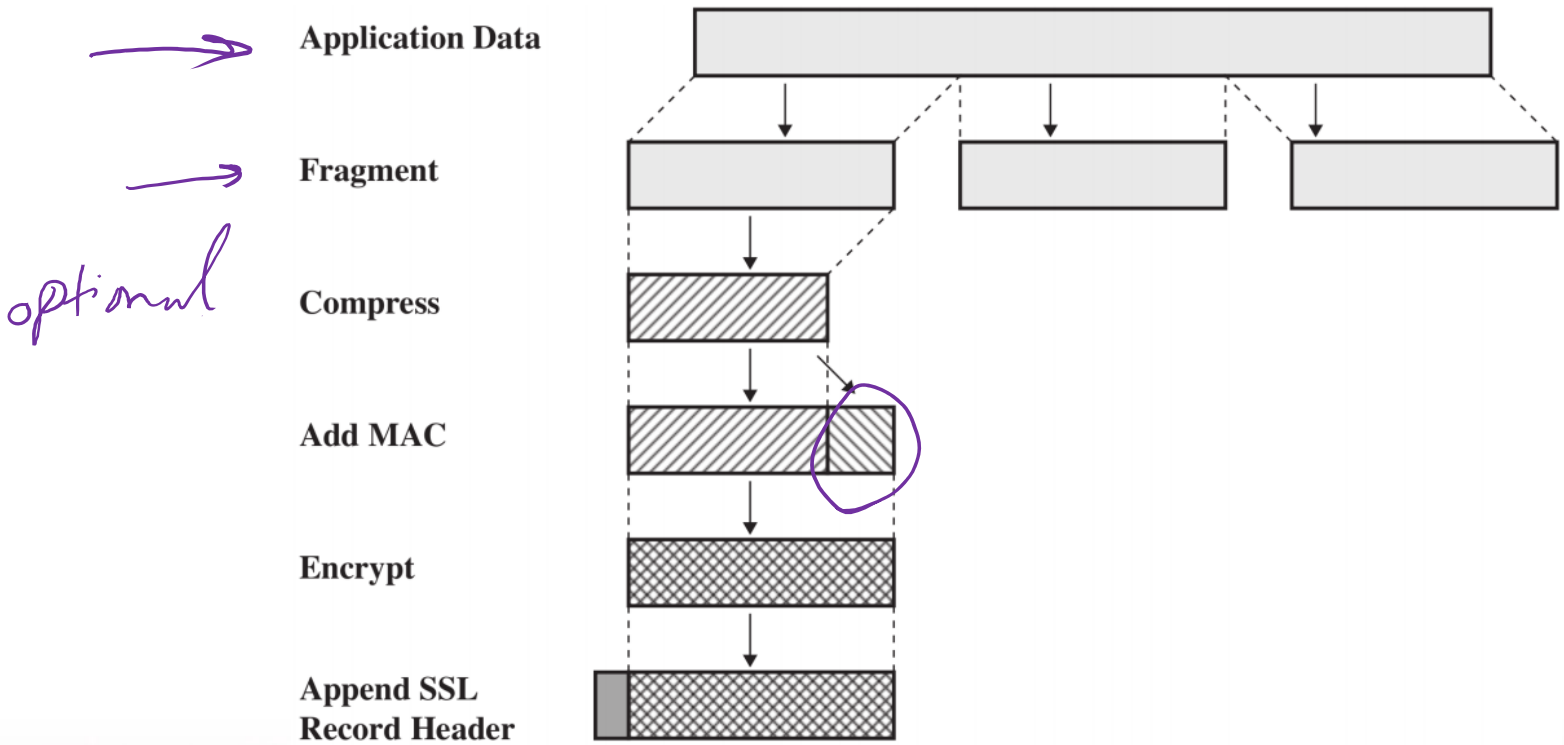
Change Cipher: change cipher for use in connection

Alert: alert peer entity of status/warning/error

Connections and Sessions

- SSL connection corresponds with TCP connection
  - Client and server may have multiple connections
- SSL session is association between client and server
  - Session created with Handshake protocol
  - Multiple connections can be associated with one session
  - Security parameters for session can be shared for connections
- State information is stored after Handshake protocol
  - Session: ID, certificate, compression, cipher spec, master secret, . . .
  - Connection: random values, encrypt keys, MAC secrets, IV, sequence numbers, . . .

SSL Record Protocol Operation



SSL Handshake Protocol

- Allow client and server to authenticate each other
- Negotiate encryption and MAC algorithms, exchange keys
  - Key Exchange: RSA, Diffie-Hellman
  - MAC: HMAC using SHA or MD5
  - Encryption: RC4, RC2, DES, 3DES, IDEA, AES
- Multiple phases:
  1. Establish security capabilities: client proposes algorithms, server selects one
  2. Server authentication and key exchange
  3. Client authentication and key exchange
  4. Finish setting up connection

```
▾Secure Sockets Layer
 ▾TLSv1.2 Record Layer: Handshake Protocol: Client Hello
  Content Type: Handshake (22)
  Version: TLS 1.0 (0x0301)
  Length: 184
  ▾Handshake Protocol: Client Hello
   Handshake Type: Client Hello (1)
   Length: 180
   Version: TLS 1.2 (0x0303)
```

SSL Handshake Protocol (cipher support)

```
▾Cipher Suites (11 suites)
  Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02
  Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
  Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a)
  Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009)
  Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)
  Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)
  Cipher Suite: TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0x0033)
  Cipher Suite: TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x0039)
  Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
  Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
  Cipher Suite: TLS_RSA_WITH_3DES_EDE_CBC_SHA (0x000a)
```

SSL Handshake Protocol (Data Encryption)

```
▾TLSv1.2 Record Layer: Application Data Protocol: http
  Content Type: Application Data (23)
  Version: TLS 1.2 (0x0303)
  Length: 329
  Encrypted Application Data: 0000000000000001999dbe1b0
```

```
0040  4b 1b 17 03 03 01 49 00  00 00 00 00 00 00 00 01 99
0050  9d be 1b 07 9e 36 1d f6  c7 c2 92 1e 41 af 95 cd
0060  7f 3b 0c 26 ed 14 e2 b1  87 76 51 b8 06 77 fb 18
0070  5e c0 e6 54 19 98 ad 0a  e4 1b d0 df c6 02 8f 0a
0080  ef a0 28 7e dd 61 b9 2f  1e da 17 7a 5c d9 e8 17
0090  d5 d0 d3 d7 71 79 74 d9  a3 e0 3d 54 d9 ab f3 c1
00a0  68 6e cb 1f ec 95 11 7a  07 05 f5 e3 44 52 35 f9
```

```
.....6
.;.&...
^..T...
..(~.a
....qyt
hn.....
5k.....
!.q....
```

This should be the get request but since it is encrypted you can’t know even what is the webpage requested!

Video summary

- How does HTTP Work?
- HTTP Security Issues
- Using HTTPS
- Introducing SSL/TLS