Web Security

CSCI 454/554

- Web now widely used by business, government, individuals
- but Internet & Web are vulnerable
- have a variety of threats
  - integrity
  - confidentiality
  - denial of service
  - authentication
- need added security mechanisms
SSL (Secure Socket Layer)

- transport layer security service
  - originally developed by Netscape (v2 in 1995)
  - SSL version 3 in 1996
  - subsequently became Internet standard known as TLS (Transport Layer Security) == SSL v3.1
- SSL (TLS) is on top of TCP
  - a shim between TCP and application protocols
  - uses TCP to provide a reliable end-to-end service
- SSL has two layers of protocols

Security Services at Different Layers

(a) Network Level
(b) Transport Level
(c) Application Level
SSL Architecture

- **SSL session**
  - an association between client & server
  - created by the Handshake Protocol
  - define a set of cryptographic parameters
  - may be shared by multiple SSL connections

- **SSL connection**
  - a transient, peer-to-peer, communications link
  - associated with one SSL session
SSL Record Protocol

- **confidentiality**
  - using symmetric encryption with a shared secret key defined by Handshake Protocol
  - AES, RC2-40, DES-40, DES, 3DES
  - message is compressed before encryption

- **message integrity**
  - using a MAC with shared secret key
  - similar to HMAC but with different padding

Record Protocol Operation

```
Application Data
  | Fragment/Combine
  |----------------
  | Record Protocol Units
  | abc def ghi
  | Compress
  |----------------
  | Compressed Unit
  | MAC
  | Encrypt
  |----------------
  | Encrypted
  | Transmit
  |----------------
  | TCP Packet
```
Record Type

- Four types of record
  - User data
  - Handshake messages
  - Alerts
    - Error messages
    - Notification of connection closure
  - Change cipher spec

Cipher Suite

- For public-key, symmetric encryption and certificate verification we need
  - public-key algorithm
  - symmetric encryption algorithm
  - message digest (hash) algorithm
- This collection is called a cipher suite
- SSL supports many different suites
- Client and server must decide on which one to use
- The client offers a choice; the server picks one
SSL Change Cipher Spec Protocol

- one of three SSL specific protocols which use the SSL Record protocol
- a single message
  - Single byte with value of 1
- causes pending state to become current
- updating the cipher suite in use

SSL Alert Protocol

- conveys SSL-related alerts to peer entity
- severity
  - warning or fatal
- specific alert
  - unexpected message, bad record mac, decompression failure, handshake failure, illegal parameter
  - close notify, no certificate, bad certificate, unsupported certificate, certificate revoked, certificate expired, certificate unknown
- compressed & encrypted like all SSL data
SSL Handshake Protocol

- allows server & client to:
  - authenticate each other
  - to negotiate encryption & MAC algorithms
  - to negotiate cryptographic keys to be used
- comprises a series of messages in phases
  - Establish Security Capabilities
  - Server Authentication and Key Exchange
  - Client Authentication and Key Exchange (optional)
  - Finish

Simplified handshake

1. Client→Server: Hello

2. Server→Client: Server’s Certificate
   Client verifies Server’s public key $K_s$

3. Client→Server: $E_{K_s}[\text{master}_\text{secret}]$

- Vulnerable to replay attacks
Enhanced handshake

1. Client → Server: *Hello*, R1
2. Server → Client: Server’s Certificate, R2
   - Client verifies Server’s public key $K_S$
3. Client → Server: $E_{K_s^+}[\text{pre_master_secret}]$
   - Master_secret = $f(R1, R2, \text{pre_master_secret})$
   - Two steps in Master_secret generation:
     - Pre_master_key is exchanged
     - Then both side compute the master_secret

---

Diagram:

- **Phase 1**: Establish security capabilities, including protocol version, session IDs, cipher suite, compression method, and initial random numbers.
- **Phase 2**: Server may send certificate, key exchange, and request certificate. Server signals end of hello message phase.
- **Phase 3**: Client sends certificate if requested. Client sends key exchange. Client may send certificate verification.
- **Phase 4**: Change cipher suite and finish handshake protocol.

Note: Shaded transfers are optional or situation-dependent messages that are not always sent.
SSL Handshake

CLIENT

- Client and Server Hello
- Certificate
  - Contains server’s public key
- Client Key Exchange
  - Pre-master secret encrypted with server’s public key
- Generate pre-master secret
- Compute master sec from pre-master secret

SERVER

- Client now has server’s public key
- Decrypt pre-master secret
- ChangeCipherSpec
- Finished
- Server’s digital certificate
- Compute master_sec from pre-master secret

Session Reuse

- Avoids public key operations
- Use master-secret to generate new session keys
**TLS (Transport Layer Security)**

- IETF standard RFC 2246 similar to SSLv3
- with minor differences
  - in record format version number
  - uses HMAC for MAC
  - a pseudo-random function expands secrets
  - has additional alert codes
  - some changes in supported ciphers
  - changes in certificate negotiations
  - changes in use of padding

**Web Application Security**

- Cookie Management
- Phishing Attacks
- Code Injection Attacks
  - SQL Injection
  - Cross-site scripting
HTTP Cookies

- HTTP – a stateless protocol
- HTTP Cookies (RFC 2109, 2965, 2964)
  - Contain server-initiated state information
  - Passed between an origin server and user agent to maintain stateful sessions and personalize webpages
- Serious privacy and security side-effects
  - Monitor browsing habits
  - Associate what you’ve looked at with who you are
  - Stolen cookies may incur serious security problems

Cookie Classification

- Based on origin and destination
  - First-party cookies
  - Third-party cookies
- Based on lifetime
  - Session cookies
  - Persistent cookies
Cookie Management

- Privacy policy options in Web browsers
  - Disabling third party cookies
  - Only allowing session cookies
  - Asking user for each cookie usage
  - Viewing or removing cookies
- Browser add-ons or standalone tools
- The main problem
  - Ignored or misused by users, or bad usability

Typical Phishing Page

Weird URL
http instead of https
Human User: the Weakest Link

- Phishers convincingly imitate the appearance of legal web sites.
- Users tend to ignore security indicators or warnings
- Users don’t necessarily interpret security cues appropriately

What is SQL Injection Attacks

- Craft and input SQL statements in a web form to get a badly designed website to dump the database content to the attacker
- User input is
  - incorrectly filtered for string literal escape characters embedded in SQL statements
  - not strongly typed and unexpectedly executed.
What is Cross-Site Scripting?

The three conditions for Cross-Site Scripting:
1. A Web application accepts user input
   - Well, which Web application doesn't?
2. The input is used to create dynamic content
   - Again, which Web application doesn't?
3. The input is insufficiently validated
   - Most Web applications don't validate sufficiently!

Scripting: Web Browsers can execute commands
- Embedded in HTML page
- Supports different languages (JavaScript, VBScript, ActiveX, etc.)
- Most prominent: JavaScript

“Cross-Site” means: Foreign script sent via server to client
- Attacker makes“ Web-Server deliver malicious script code
- Malicious script is executed in Client’s Web Browser

Attack:
- Steal Access Credentials, Denial-of-Service, etc.
- Execute any command at the client machine
Defense Against Code Injections

- Validate input
- Limit program functionality
  - Don’t leave open ended-functionality
- Execute with limited privileges

Summary

- have considered:
  - need for web security
  - SSL/TLS transport layer security protocols
  - Web Application Security Problems