

Web Development using Java, JSP, and Web Services

Web Security

Today

Security

Threats

Attacks

Web Security

Cryptography

Public Key Infrastructures

HTTPS

Next Time

# Web Development using Java, JSP, and Web Services

## Web Security

### Lecture #8 2008

Web Development using Java, JSP, and Web Services

Web Security

Today

Security

Threats

Attacks

Web Security

Cryptography

Public Key Infrastructures

HTTPS

Next Time

# 1 Security

## Threats

### Attacks

# 2 Web Security

## Cryptography

### Public Key Infrastructures

### HTTPS

Web Development using Java, JSP, and Web Services

Web Security

Today

Security

Threats

Attacks

Web Security

Cryptography

Public Key Infrastructures

HTTPS

Next Time

# Security

- Freedom from danger, risk, etc.; safety.
- Something that secures or makes safe; protection; defense.
- Precautions taken to guard against crime, attack, sabotage, espionage, etc.
- An assurance; guarantee.
- *Archaic.* overconfidence; cockiness.

(definitions from *dictionary.com*)

Web Development using Java, JSP, and Web Services

Web Security

Today

Security

Threats

Attacks

Web Security

Cryptography

Public Key Infrastructures

HTTPS

Next Time

# Security

- Physical security - protecting the infrastructure, e.g.
  - fire alarms
  - locked doors
- Data security - protecting the content, e.g.
  - privacy
  - integrity
- Network security - protecting the access, e.g.
  - firewalls
  - encryption

Web Development using Java, JSP, and Web Services

Web Security

Today

Security

Threats

Attacks

Web Security

Cryptography

Public Key Infrastructures

HTTPS

Next Time

# Security Goals

- Prevention (may hinder system availability)
- Detection (does not prevent system compromise)
- Recovery
  - stop attack
  - assess damage
  - repair damage (complex)
  - retaliate (?)

Web Development using Java, JSP, and Web Services

Web Security

Today

Security

Threats

Attacks

Web Security

Cryptography

Public Key Infrastructures

HTTPS

Next Time

# Security in layers

- System security is the sum of component securities
- Designing security in many levels increases the effort required to attack the system
- Attackers choose targets based on a risk-reward-effort analysis

Web Development using Java, JSP, and Web Services

Web Security

Today

Security

Threats

Attacks

Web Security

Cryptography

Public Key Infrastructures

HTTPS

Next Time

## Who, What, and Why

- Perpetrators
  - single and groups of hackers
  - organized crime
  - military organizations
- Targets
  - individual end-users
  - organizations and companies
  - infrastructure
- Goals
  - fame
  - financial assets
  - botnet access

◀ ▶ ⏪ ⏩ 🔍 🔄

Web Development using Java, JSP, and Web Services

Web Security

Today

Security

Threats

Attacks

Web Security

Cryptography

Public Key Infrastructures

HTTPS

Next Time

## Threats

- Snooping (disclosure)
  - Confidentiality
- Modification, alteration (deception, usurpation)
  - Integrity
- Masquerading, spoofing (deception, usurpation)
  - Integrity
- Repudiation (deception)
  - Integrity, non-repudiation
- Denial of receipt (deception)
  - Integrity, availability
- Delay (usurpation, deception)
  - Availability
- Denial of service
  - Availability

◀ ▶ ⏪ ⏩ 🔍 🔄

Web Development using Java, JSP, and Web Services

Web Security

Today

Security

Threats

Attacks

Web Security

Cryptography

Public Key Infrastructures

HTTPS

Next Time

## Threat Analysis

- Level of protection
  - Physical security
  - Psychological security (awareness, knowledge etc)
  - Virtual security
- Trust models
- System life cycles (do we need to protect old systems?)
- Detection
- Reaction

◀ ▶ ⏪ ⏩ 🔍 🔄

Web Development using Java, JSP, and Web Services

Web Security

Today

Security

Threats

Attacks

Web Security

Cryptography

Public Key Infrastructures

HTTPS

Next Time

## Threat Analysis

- Protect against the greatest risks
- Value of protected items?
- Loss expectancy (immediate and annual)
- Attack trees
- Failure Modes and Effect Analysis (FMEA)
  - Bottom up
  - How does component failure affect the system?
- Combine as a matrix

◀ ▶ ⏪ ⏩ 🔍 🔄

Web Development using Java, JSP, and Web Services

Web Security

Today

Security

Threats

Attacks

Web Security

Cryptography

Public Key Infrastructures

HTTPS

Next Time

## Attack types

- Passive: hard to detect & hard to prevent
  - surveillance
  - publication of message contents
  - traffic analysis
- Active: easier to detect, hard to prevent (alt. detect and recover)
  - masquerade
  - replay
  - alteration of message content
  - denial of service

◀ ▶ ⏪ ⏩ 🔍 🔄

Web Development using Java, JSP, and Web Services

Web Security

Today

Security

Threats

Attacks

Web Security

Cryptography

Public Key Infrastructures

HTTPS

Next Time

## (Active) Attack Methodology

- 1 Identify target
- 2 Gather information
- 3 Analyze information / locate vulnerabilities
- 4 Gain access
- 5 Execute attack
- 6 Erase attack traces

Usually enough to stop one of the above

◀ ▶ ⏪ ⏩ 🔍 🔄

Web Development using Java, JSP, and Web Services

Web Security

Today

Security  
Threats  
**Attacks**

Web Security  
Cryptography  
Public Key Infrastructures  
HTTPS

Next Time

## Security Exploits

- System vulnerabilities are exploited in attacks
- Aims to escalate privileges or take control
- Very common in large systems
- Attacks are mechanized and shared in communities
- Known security vulnerabilities are published and addressed

◀ ▶ ⏪ ⏩ 🔍 🔄

Web Development using Java, JSP, and Web Services

Web Security

Today

Security  
Threats  
**Attacks**

Web Security  
Cryptography  
Public Key Infrastructures  
HTTPS

Next Time

## Security Exploit Examples

- Malicious ActiveX controls
- Scripts targeting web browser bugs
- Malicious code in codecs
- Malicious code in images (GDI+)
- Automatic programs probing network services

◀ ▶ ⏪ ⏩ 🔍 🔄

Web Development using Java, JSP, and Web Services

Web Security

Today

Security  
Threats  
**Attacks**

Web Security  
Cryptography  
Public Key Infrastructures  
HTTPS

Next Time

## Buffer Overflow

- Data are stored beyond buffer boundaries
- The extra data overwrites adjacent memory locations
- Overflows can cause
  - changed program behavior
  - crashes
  - memory exceptions
  - usurpation of process...
- Countered by bounds checking (automatic in Java)

◀ ▶ ⏪ ⏩ 🔍 🔄

Web Development using Java, JSP, and Web Services

Web Security

Today

Security  
Threats  
**Attacks**

Web Security  
Cryptography  
Public Key Infrastructures  
HTTPS

Next Time

## SQL Injection

- Addresses vulnerabilities in database access layers
- Targets unescaped data literals or weak type access
- Injects an SQL snippet within regular SQL commands
- Countered by data filtering

◀ ▶ ⏪ ⏩ 🔍 🔄

Web Development using Java, JSP, and Web Services

Web Security

Today

Security  
Threats  
**Attacks**

Web Security  
Cryptography  
Public Key Infrastructures  
HTTPS

Next Time

## Man In The Middle

- A form of active eavesdropping
- An attacker places himself between a two parties, assuming the identity of each and relays messages
- Technically advanced and hard to detect
- Countered by (correct) use of authentication and key exchange protocols and infrastructures

◀ ▶ ⏪ ⏩ 🔍 🔄

Web Development using Java, JSP, and Web Services

Web Security

Today

Security  
Threats  
**Attacks**

Web Security  
Cryptography  
Public Key Infrastructures  
HTTPS

Next Time

## Spoofing

- A system masquerades as a target system
- A distributed form of a Trojan horse
- Web versions used for phishing user data
- Countered by raising user awareness

◀ ▶ ⏪ ⏩ 🔍 🔄

Web Development using Java, JSP, and Web Services

Web Security

Today

Security  
Threats  
Attacks

Web Security  
Cryptography  
Public Key Infrastructures  
HTTPS

Next Time

## Denial of Service

- Attacks the availability of a system
- Systems are overloaded to stop access to them
- Often performed from distributed botnets (DDOS)
- Countered by sound system design, firewalls, and redundancy in system infrastructure

◀ ▶ ⏪ ⏩ 🔍

Web Development using Java, JSP, and Web Services

Web Security

Today

Security  
Threats  
Attacks

Web Security  
Cryptography  
Public Key Infrastructures  
HTTPS

Next Time

## Web Security

- Based on cryptography
- SSL / TLS current encryption standards
- HTTPS = HTTP through a SSL tunnel (no changes in JSP required)

◀ ▶ ⏪ ⏩ 🔍

Web Development using Java, JSP, and Web Services

Web Security

Today

Security  
Threats  
Attacks

Web Security  
Cryptography  
Public Key Infrastructures  
HTTPS

Next Time

## Cryptography

- Mathematical tools for enabling trust
- Based on fundamental assumptions
  - algorithms are safe (there are no shortcuts)
  - parameter space searches for keys takes a long time
  - techniques used as intended
- Message: data
- Algorithm: the encryption method
- Key: encryption key, parameter to encryption algorithm
- Cipher text: the encrypted message

◀ ▶ ⏪ ⏩ 🔍

Web Development using Java, JSP, and Web Services

Web Security

Today

Security  
Threats  
Attacks

Web Security  
Cryptography  
Public Key Infrastructures  
HTTPS

Next Time

## One-Way Encryption

- Messages are encrypted using secret keys
- Messages can not be decrypted
- Cipher texts are (to a high probability) uniquely mapped to message content
- Cipher texts are used instead of messages in situations where messages must be kept secret (e.g., passwords)
- Closely related to hashcodes and Message Authentication Codes (MACs)

◀ ▶ ⏪ ⏩ 🔍

Web Development using Java, JSP, and Web Services

Web Security

Today

Security  
Threats  
Attacks

Web Security  
Cryptography  
Public Key Infrastructures  
HTTPS

Next Time

## Symmetric Encryption

- Commonly referred to as *private key encryption*
- Messages are encrypted and decrypted using the same key
- Anyone with access to the key can decrypt the message
- Fast
- Suffers from *the key distribution problem*

◀ ▶ ⏪ ⏩ 🔍

Web Development using Java, JSP, and Web Services

Web Security

Today

Security  
Threats  
Attacks

Web Security  
Cryptography  
Public Key Infrastructures  
HTTPS

Next Time

## Asymmetric Encryption

- Commonly referred to as *public key encryption*
- Messages are encrypted using key pairs (public & private)
- One key used for encryption, the other for decryption
- Public key distributed as much as possible
- Private key kept secret
- Versatile and more secure than symmetric algorithms
- Slow

◀ ▶ ⏪ ⏩ 🔍

## Asymmetric Encryption

- Encrypt message using public key - encryption
- Encrypt message using private key - signatures
- Messages can be both encrypted and signed
- As long as the keys can be trusted
  - messages can be kept secret (only receiver can decrypt)
  - senders and receivers can be authenticated
  - message content can be trusted

## Certificates

- Certificate = signed tuple of public key & identity
- Certificates can be self-signed or signed by others
- Self-signed certificates can be used for encryption (but suffer from *the key distribution problem*)
- Certificates signed by trusted parties can be used for encryption, authentication and message integrity checks

## Public Key Infrastructures (PKI)

- Virtual infrastructures consisting of clients, servers and Certificate Authorities (CA)
- CAs are trusted third parties which provide signed certificates (i.e., signs public keys)
- CA certificates are distributed in browsers and similar tools (trusted and considered known by all)
- Since CA public keys are known, (signed) certificates can be validated offline (without connecting to the CA)
- Secure connections are established between parties using certificates and encryption algorithms
- Network traffic *tunneled* through encrypted channels

## Secure Socket Layer (SSL)

- A protocol for establishing secure connections using certificates and cryptography algorithms
- Transport Level Security (TLS) = SSL v3.0 (almost)
- Clients use server certificate to authenticate server
- Servers use client certificate to authenticate client (optional)
- Once identities have been established, encryption keys are exchanged and symmetric encryption algorithms are used
- SSL clients uses *keystores* to manage certificates and keys

## Secure Socket Layer (SSL)

Bruce Schneier, *Secrets and Lies* (page 239):  
*"As it is used, with the average user not bothering to verify the certificates and no revocation mechanism, SSL is simply a (very slow) Diffie-Hellman key-exchange method. Digital certificates provide no actual security for electronic commerce: it's a complete sham."*

## Keystores

- An encrypted database used to store keys and certificates
- Usually stored in a single file called `.keystore`
- Applications must provide database decryption key (username & password) to access keystore content
- Keystores only containing public keys and certificates are commonly referred to as *truststores*
- Keystores can be shared between SSL applications (usually only done for truststores)

Web Development using Java, JSP, and Web Services

Web Security

Today


Security Threats Attacks

Web Security Cryptography Public Key Infrastructures **HTTPS**

Next Time

## HTTPS

- Not an actual protocol
- HTTPS = HTTP through a SSL/TLS tunnel
- The server needs to be provided with a certificate
- If the server is to authenticate clients, the clients need (CA signed) certificates as well
- HTTPS Web servers usually references keystores via configuration (providing filename, username, password)
- Default port 443 (HTTP default port is 80)
- JSP can check if a page was requested via HTTPS using `request.isSecure()`
- HTTPS / SSL is considered safe (today)



Web Development using Java, JSP, and Web Services

Web Security

Today


Security Threats Attacks

Web Security Cryptography Public Key Infrastructures **HTTPS**

Next Time

## Summary

- Security is a process, not a product
- Cryptography is the tool for web security
- No changes in JSP required to use HTTPS (web server reconfiguration may be required)
- Web server needs a certificate
- JSP can require clients to use HTTPS



Web Development using Java, JSP, and Web Services

Web Security

Today

Security Threats Attacks

Web Security Cryptography Public Key Infrastructures **HTTPS**

Next Time

## Next Time

- XML & XML Schema

