CPET 499/ITC 250 Web Systems

Chapter 16
Security

Text Book:
*** Fundamentals of Web Development, 2015, by Randy Connolly and Ricardo Hoar, published by Pearson**

Paul I-Hai Lin, Professor
http://www.etcs.ipfw.edu/~lin

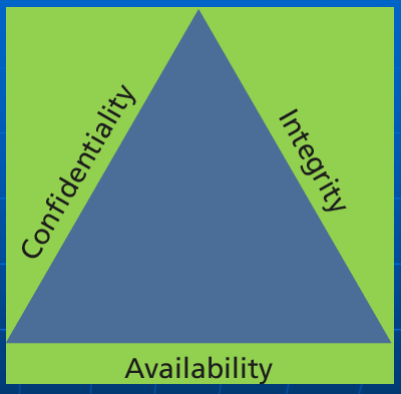
Topics

Chapter Objectives

- A wide range of security principles and practices
- Best practices of authentication systems and data storage
- About public key cryptography, SSL, and certificates
- How to proactively protect your site against common attacks

Security Principles

- Information Security
- The CIA Triad (Figure 16.1)
  - **Confidentiality – The principle of maintaining privacy for the data you are storing, transmitting, etc**
  - **Integrity – The principle of ensuring that data is accurate and correct.**
  - **Availability – The principle of making information available when needed to authorized people.**
- **Security Standards**
  - **ISO standards ISO/IEC 27002-27--37**



Risk Assessment and Management

- **Risk – a measure of how likely an attack is, and how costly the impact of the attack would be if successful**
- **Security Standards – ISO/IEC 27002-270037**
- **Actors, Impacts, Threats, and Vulnerability**
- **Actors**
  - **Internal actors**
  - **External actors**
  - **Partner actors**
- **Impacts**
  - **A loss of availability**
  - **A loss of confidentiality**
  - **A loss of integrity**

Risk Assessment and Management

- **Threats**
  - Refers to a particular path that a hacker cloud use to exploit a vulnerability and gain unauthorized access to your system.
  - Also called attack vectors
- Categories of Threats (STRIDE)
  - Spoofing – use someone else's info to access the system
  - Tampering – modify some data in unauthorized ways
  - Repudiation – remove all trace of their attack, so they cannot be held accountable for other damage done
  - Information disclosure – access data they should bot be able to
  - Denial of service – prevent the real users from accessing the systems
  - Elevation of privilege

Risk Assessment and Management

- Vulnerability – the security holes in your system
- The top 10 classes of vulnerability from the Open Web Application Security Project (2013): https://www.owasp.org/index.php/Top_10_2013-Top_10
  - A1. Injection
  - A2. Broken authentication and session management
  - A3. Cross-site scripting
  - A4. Insecure direct object reference
  - A5. Security misconfiguration
  - A6. Sensitive data exposure
  - A7. Missing function level access control
  - A8. Cross-site request forgery (CSRF)
  - A9. Using components with unknown vulnerabilities
  - A10. Un-validated redirects and forwards

Assessing Risk

- NIST Risk Management Guide for Information Technology Systems, http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf
- Table 16.1 Examples an Impact/Probability Risk Assessment Table Using 16 as the Threshold

Probability \ Impact(n^2)	Very Low	Low	Medium	High	Very High
Very High	5	10	20	40	80
High	4	8	16	32	64
Medium	3	6	12	24	48
Low	2	4	8	16	32
Very low	1	2	4	8	16

Assessing Risk

- Table 16.1 Examples an Impact/Probability Risk Assessment Table Using 16 as the Threshold

Probability \ Impact (n^2)	Very low	Low	Medium	High	Very high
Very high	5	10	20	40	80
HIgh	4	8	16	32	64
Medium	3	6	12	24	48
Low	2	4	8	16	32
Very low	1	2	4	8	16

Security Policy

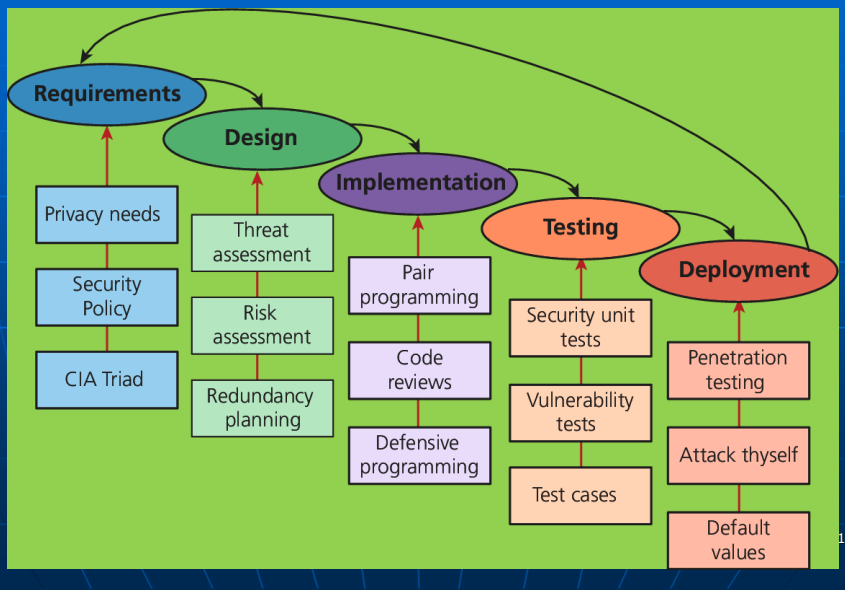
- Usage Policy
  - Social networking policy at work?
- Authentication Policy
  - Access badge
  - Biometric ID
  - Password
  - VPN
- Legal Policy

Business Continuity & Plans

- Admin Password Management
- Backups and Redundancy
- Geographic Redundancy
- Storage Mock Events
- Auditing

Security By Design

Figure 16.2 Some examples of security input into the SDLC



Security By Design

- **Code Reviews**
  - Peer-reviewed before committing it to the repository
  - Company coding style and practice
  - Informal and formal review process
- **Unit Testing**
  - Code Modules
  - Class
  - Security holes
- **Pair Programming**
  - Two programmers working together
- **Security Testing**
  - Testing the system against scenarios that attempt to break the final system
  - Penetration testing
- **Secure by Default**

Social Engineering

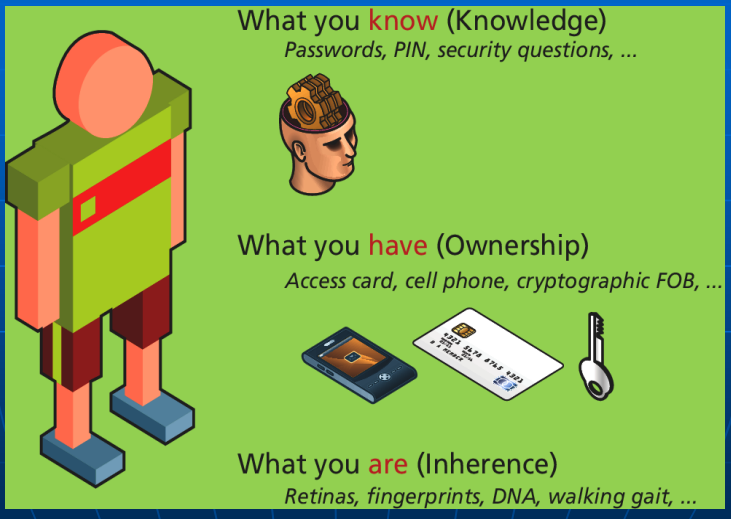
- Social engineering
  - A broad term given to describe the manipulation of attitudes and behaviors of a populace, often through government or industrial propaganda and/or coercion.
  - A human part of information security that increases the effectiveness of an attack.
  - Social Engineering (Security), https://en.wikipedia.org/wiki/Social_engineering_(security)
  - http://www.social-engineer.org/
- Two popular techniques
  - Phishing scams
  - Security theater

Social Engineering

- Other References
  - Social Engineering (Security), https://en.wikipedia.org/wiki/Social_engineering_(security)
  - http://www.social-engineer.org/
- Top 5 Social Engineering Exploit Techniques, by James Heary, Network World, http://www.pcworld.com/article/182180/top_5_social_engineering_exploit_techniques.html
  - 1) Familiarity exploit
  - 2) Creating a hostile situation
  - 3) Gathering and using information
  - 4) Get a job there
  - 5) Reading body language

Authentication

Figure 16.3 Authentication Factors



Authentication

- **Authentication Factors**
  - Knowledge factors: password, PIN, challenge questions
  - Ownership factors: driver license, passport, cell phone, key to a lock
  - Inherence factors: biometric data – fingerprints, retinal patterns, DNA sequence
- **Single-Factor Authentication**
  - Password/ Magnetized key badge
- **Multi-Factor Authentication**
  - ATM Machine: Access card and PIN
- **Third-Party Authentication**
  - Open Authentication (OAuth)

Third Party Authentication

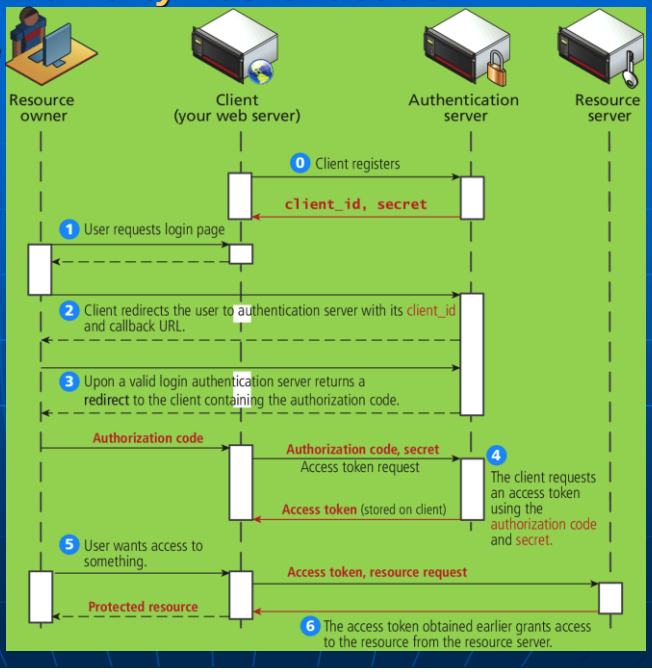
- **Open Authentication (OAuth), http://oauth.net/**
  - A open protocol to allow secure authorization in a simple standard method from web, mobile and desktop applications.
  - This specification is likely to produce a wide range of non-interoperable implementation.
  - OAuth 2.0, http://oauth.net/2/, Client and Server Libraries for Java, PHP, Python, NodeJS, Ruby, .NET, etc
  - Four Roles: Resource owner, Resource server, Client, Authorization server

Third Party Authentication

- **Open Authentication (OAuth), http://oauth.net/**
  - Four Roles
    - Resource owner – normally the end user who can gain access to the resource
    - Resource server – host the resources and can process request using access tokens
    - Client – the application making requests on behalf of the resource owner
    - Authorization server – issues tokens to the client upon successful authentication of the resource owner. (often this is the same as the resource server)

Third-Party Authentication

- Figure 16.4 The Steps required to register and authenticate a user using OAuth



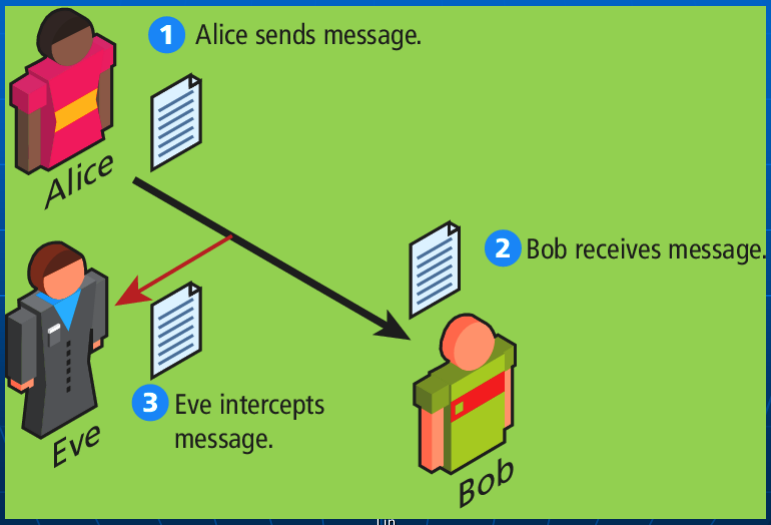
Authorization

Some examples in web development where proper authorization increases security

- Using a separate database user for read/write privileges on a database
- Providing each user an account where they can access their own file securely
- Setting proper Read/Write/Execute permissions
- Ensuring Apache is not running as the root account (an account that can access everything)

Cryptography

Figure 16.4 Message Intercepting

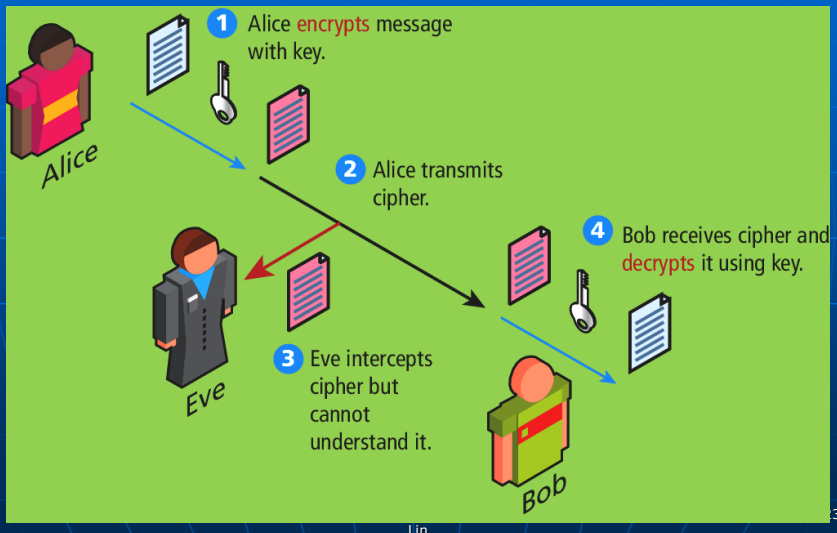


Cryptography

- Cipher – a message that is scrambled so that it cannot easily be read, unless one has some secrete key
- Key – Can be a “number”, “phrase”, “page from a book”
- Encryption
- Decryption

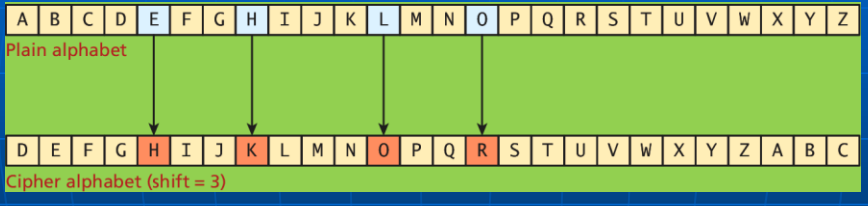
Cryptography

Figure 16.5 Symmetric encryption



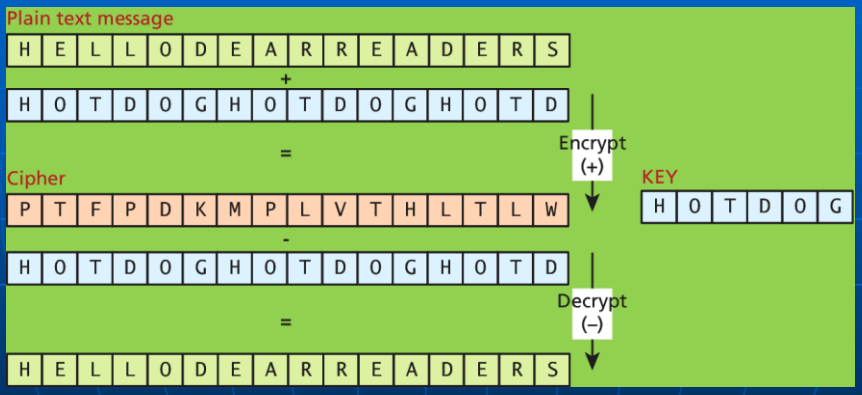
Substitution Ciphers – Cesar Cipher

- Figure 16.7 Caser Cipher for shift value of 3 (Hello => KHOOR)



Substitution Ciphers – Vigenere

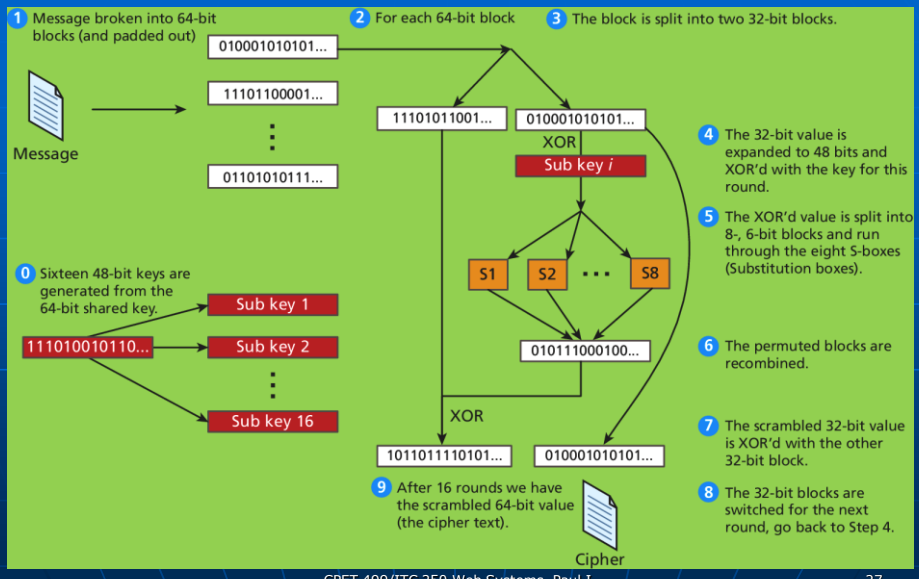
- Figure 16.7 Vigenere cipher example with key “hotdog”



Substitution Ciphers

- One-time Pad Cipher
- Modern Block Ciphers
  - Scrambled 64 or 128 bits block as a time
  - Data Encryption Standard (DES)
  - Advanced Encryption Standard (AES)

Figure 16.10 High-level illustration of the EDF cipher



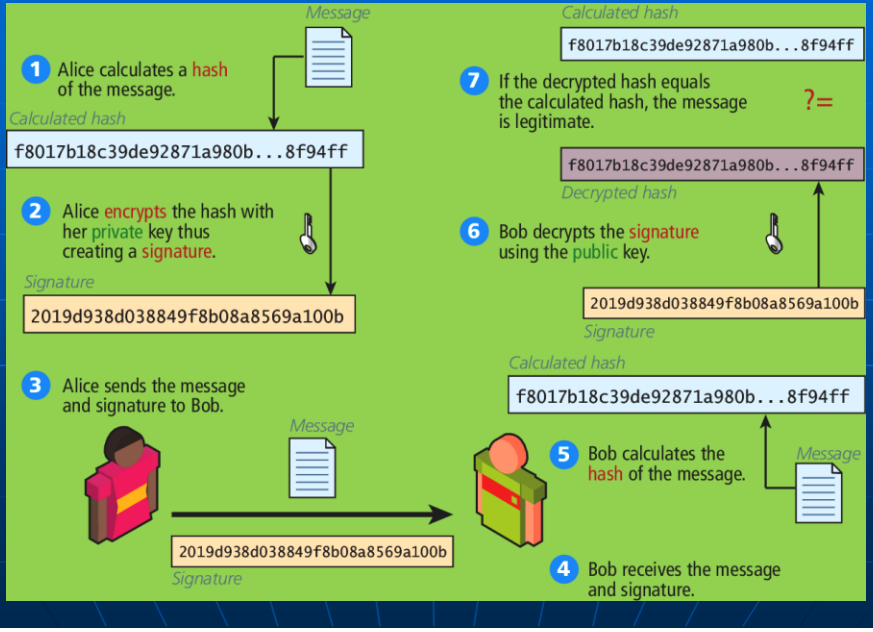
Public Key Cryptography

- **Public key cryptography (asymmetric cryptography)**
- **Using two distinct keys:**
  - A public key – widely distributed
  - A private key
- **Diffie-Hellman Key Exchange algorithm**
- **RSA (Ron Rivest, Adi Shamir and Leonard Adeleman) algorithm underpinning the HTTPs protocol**

Digital Signatures

- **A mathematically secure way of validating that a particular digital document**
  - was created by the person claiming to create it (authenticity)
  - was not modified in transit (integrity), and
  - cannot be denied (non-repudiation)
- **An example using public key cryptography**

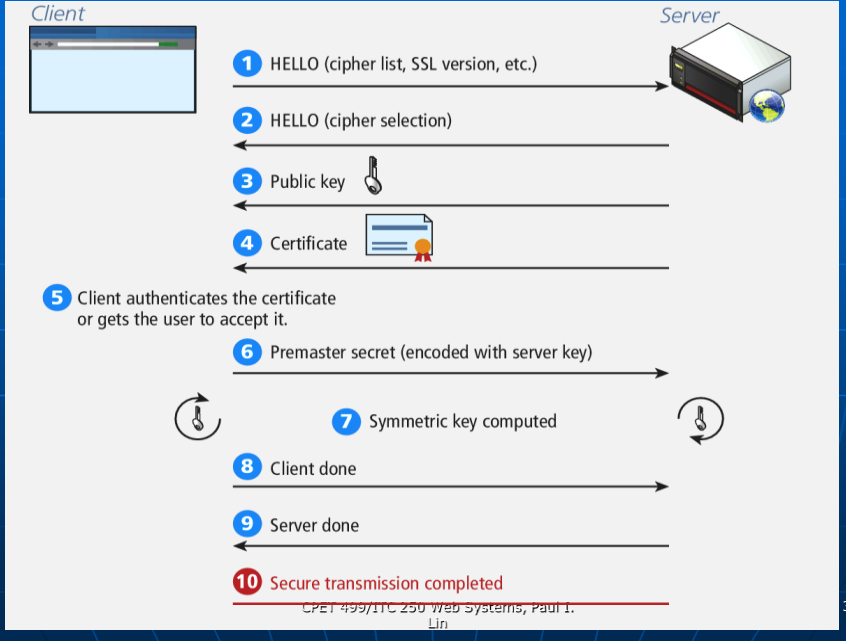
Digital Signatures
Figure 16.12 Digital Signature and Validation



Hypertext Transfer Protocol Secure (HTTPs)

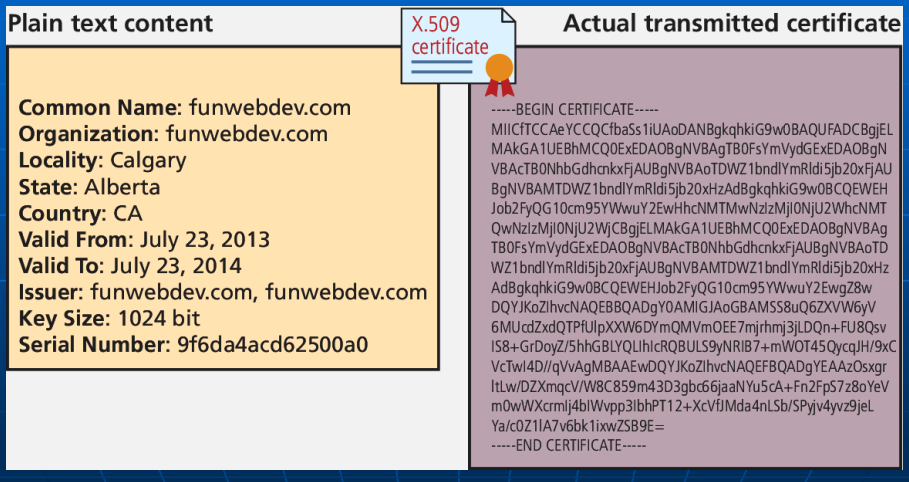
- HTTPs is the HTTP running on top of the Transport Layer Security (TLS)
- TLS v1.0 – an improvement on Secure Socket Layer 3.0 (SSL)
- For compatibility reason, we refer it as HTTP running on TLS/SSL
- Secure Handshakes
- Certificates and Authorities
  - Self-signed Certificates

Figure 16.14 SSL Secure Handshake



Certificates and Authorities

- Figure 16.15 The content of a self-signed certificate for funwebdev.com (X.509 certificate Example)

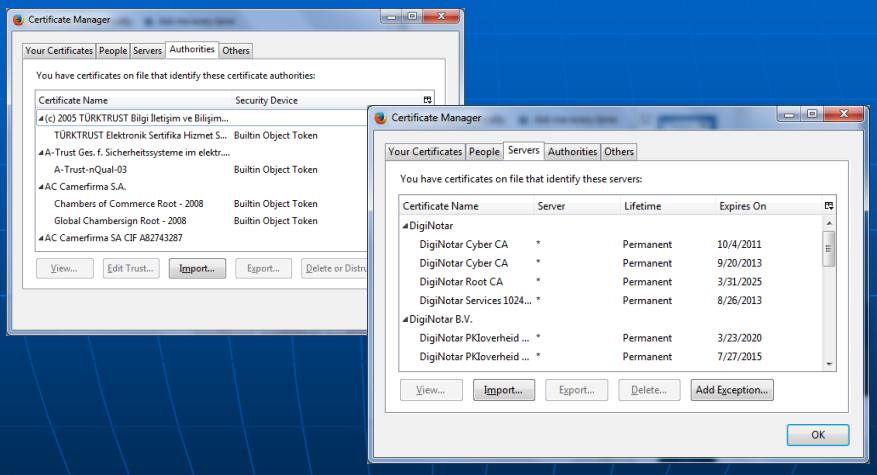


Certificates and Authorities

- Certificate - X.509 certificate which contains many details including
  - Algorithm used
  - The domain it was issued for
  - Some public key information
- X.509 Client Certificate, https://help.sap.com/saphelp_nw73/helpdata/en/43/dc1fa58048070ee10000000a422035/content.htm
- X.509 Certificate Tool, https://msdn.microsoft.com/en-us/library/aa529278.aspx
- X.509 Certificates and Certificate Revocation Lists (CRLs), http://docs.oracle.com/javase/7/docs/technotes/guides/security/cert3.html

Firefox Certificate Management Interface

- Options => Certificates => View Certificates (Some examples)



Security Best Practices

- **Data Storage**
  - Secure Hash
  - Salting the Hash
- **Monitor Your Systems**
  - System Monitors
  - Access Monitors
  - Automate Intrusion Blocking
- **Audit and Attack Thyself**

Security Best Practices – Linux Systems

References

- **Ch. 15 Security, *Linux System Administration,* Linux System Administration, 2nd ed, by Vicki Stanfield and Roderick Smith, published by Sybex**
- **Ch. 15 System Security, *A Practical Guide to Ubuntu Linux*, by Mark G. Sobell, 4th edition, published by Prentice Hall**
- **Password Formats - Basic Authentications, https://httpd.apache.org/docs/2.2/misc/password_encryptions.html**
- **The apache-md5 package (OpenSSL MD5() function), https://hackage.haskell.org/package/apache-md5**

Security Best Practices – Microsoft Systems and Servers

References

- **Windows 7: Security Features**, http://www.microsoft.com/security/pc-security/windows7.aspx
- **Windows 10 Security Overview**, https://technet.microsoft.com/en-us/library/mt601297(v=vs.85).aspx
- **What's New in Windows Server 2016 Technical Preview**, Aug. 18, 2015, https://technet.microsoft.com/en-us/library/dn765472.aspx
- **Security Best Practice for IIS 8**, June 24, 2013, https://technet.microsoft.com/en-us/library/jj635855.aspx
- **Windows Server**, https://technet.microsoft.com/en-us/library/bb625087.aspx

Security Best Practices – Linux Systems

Reference - Linux System Administration, 2nd ed, by Vicki Stanfield and Roderick Smith, published by Sybex

- User-based Security
- Port Security
- Host-based Security
- Physical Access Security
- File and/or Device Assignment of Permission

Security Best Practices – Linux Systems

Reference - Linux System Administration, 2nd ed, by Vicki Stanfield and Roderick Smith, published by Sybex

User-based Security:

- What resources should be available to the claimed user at this time?
- Pluggable Authentication Modules (PAM) to secure the system from intrusion by unauthorized users.
- Password Authentication Algorithms
  - DES (Data Encryption Standard) – encoded using the Federal Data Encryption standard algorithm
  - MD5 (Message Digest Algorithm, version 5) –
    - Uses RSA Data Security, Inc's algorithm
    - By default on most Linux system

Security Best Practices – Linux Systems

Reference - Linux System Administration, 2nd ed, by Vicki Stanfield and Roderick Smith, published by Sybex

User-based Security: Hashing Passwords

- Creating Password
  - Salt (2-character) + Clear Text Password => [Hashing Algorithm] => Salt/Password Hash
- Logging In
  - (User Supplied Password) + (/etc/shadow or /etc/passwd) Salt => [Hashing Algorithm] => Hash + Stored Hash (/etc/shadow or /etc/password) => Login Fail (Not equal to) OR Login Succeeds (Equal to)

Security Best Practices – Linux Systems

Reference - Linux System Administration, 2nd ed, by Vicki Stanfield and Roderick Smith, published by Sybex

- **User-based Security**:
  - What resources should be available to the claimed user at this time?
  - Pluggable Authentication Modules (PAM) to secure the system from intrusion by unauthorized users.
- **Port Security**:
  - Protect network ports from unauthorized hosts and networks
  - Handled by the kernel
  - IP firewall administration (IP chains or IP tables)
- **Host-based Security**:
  - Restrict network access to system resources and services based on the requesting hosts.

Security Best Practices – Linux Systems

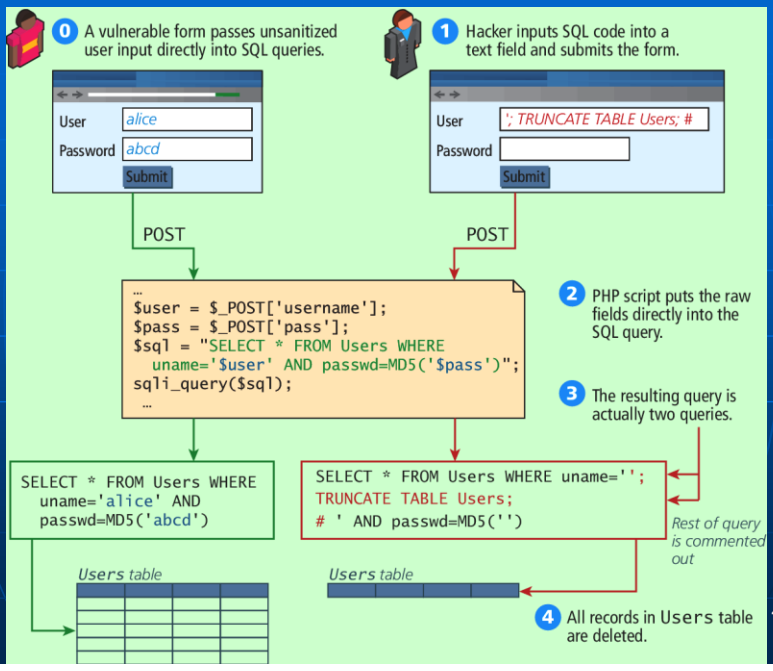
Reference - Linux System Administration, 2nd ed, by Vicki Stanfield and Roderick Smith, published by Sybex

- **User-based Security:**
  - What resources should be available to the claimed user at this time?
  - Pluggable Authentication Modules (PAM) to secure the system from intrusion by unauthorized users.
- **Port Security:**
  - Protect network ports from unauthorized hosts and networks
  - Handled by the kernel
  - IP firewall administration (IP chains or IP tables)
- **Host-based Scurity:**
  - Restrict network access to system resources and services based on the requesting hosts.

Common Threat Vectors

- SQL Injection
  - The attack technique of using reserved SQL symbol to try and make the web server execute a malicious query other than what was intended.
  - Must Sanitize inputs
  - Give Least possible privileges
- Cross-Site Scripting (XSS)
- Insecure Direct Object Reference
- Denial of Service
- Security Misconfiguration

Figure 16.19 a SQL Injection attack (right) and intended usage (left)

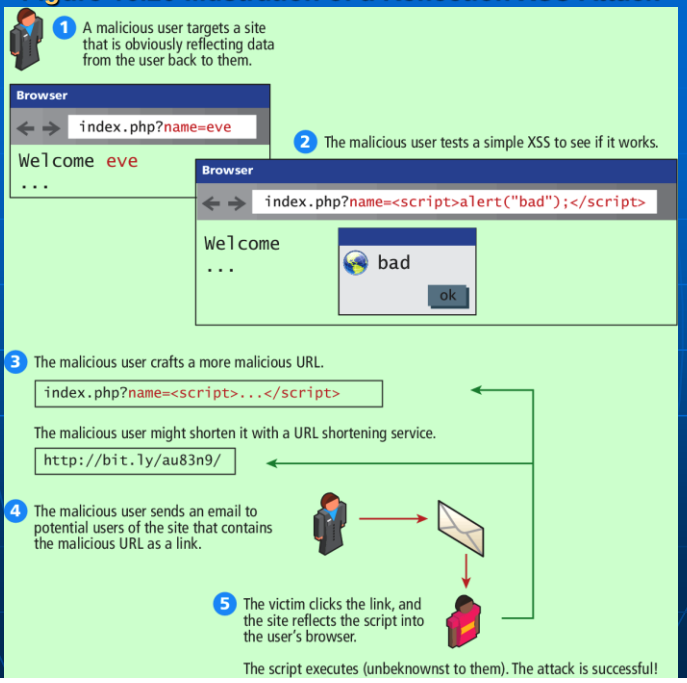


45

Cross-Site Scripting

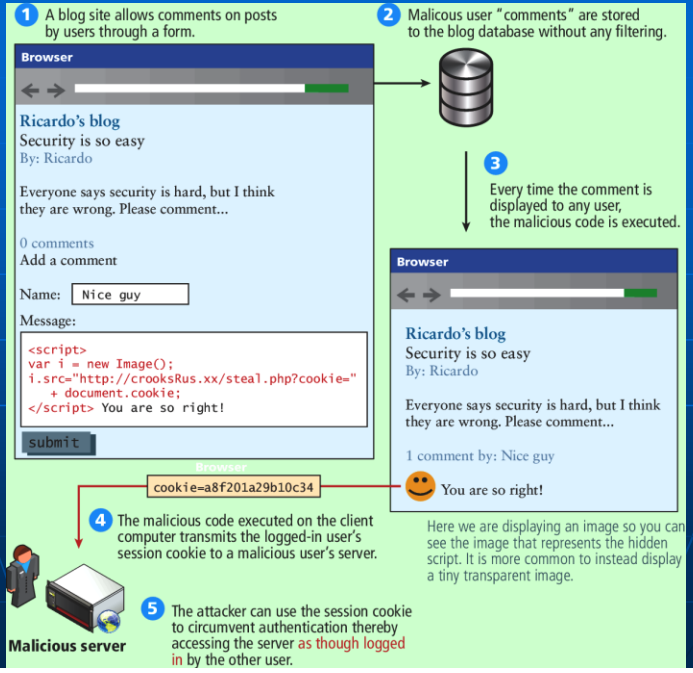
- **Cross-Site Scripting (XSS) refers to a type of attack in which a malicious script (JavaScript, VBScript, or Action Script, etc) is embedded into an otherwise trustworthy website.**
- **Two main categories of XSS**
  - **Reflected XSS** (Non-persistent XSS)
    - Are attacks that send malicious content to the sever, so that in the server response, the malicious content is embedded
  - **Store XSS** (Persistent XSS)
    - More dangerous which may impacts all users visit the site

Figure 16.20 Illustration of a Reflection XSS Attack



47

Figure 16.21 Illustration of a Stored XSS Attack



48

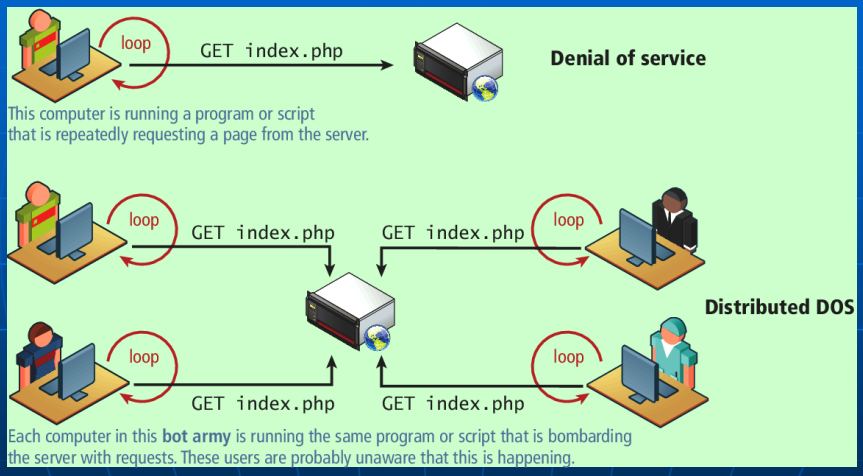
Common Threat Vectors

- **Insecure Direct Object Reference**
  - **Expose some internal value or key of the application to the user**
  - **Then the attackers can then manipulate the internal keys to gain access to things that should not have access to**
  - **Examples:**
    - An archive of the site's PHP code or passwords can be potentially accessed or downloaded
    - A database key in the URLs that are visible to users
    - Storing files on the server
- **Denial of Service**
- **Security Misconfiguration**

Denial of Services

- **Denial of Service attacks (DoS)**
  - are attacks that aim to overload a server with illegitimate requests in order to prevent the site from responding to the legitimate ones,
  - Methods of prevention
    - Blocking the IP address in the firewall or the Apache server
- **Distributed DoS Attack (DDoS)**
  - Attacks are coming from multiple machines
  - Recent DDoS attack on Spamhaus servers (generates 300 Gbps worth of requests), http://www.spamhaus.org/news/article/695/answers-about-recent-ddos-attack-on-spamhaus

Figure 16.22 DoS and DDoS



Security Misconfiguration

- **Out-of-Date Software**
- **Open Mail Relays**
  - Refers to any email server that allows someone to route email through without authentication
- **More Input Attacks**
  - Refers to the potential vulnerability that occurs when the users through their HTT requests, transmit a variety of strings and data that are directly used by the server **without sanitation**.
- **Virtual Open Mail Relay – Figure 14.23**
  - HTML web email send to any email addresses
- **Arbitrary program execution – Figure 16.24**

Figure 16.23 Virtual open relay exploit

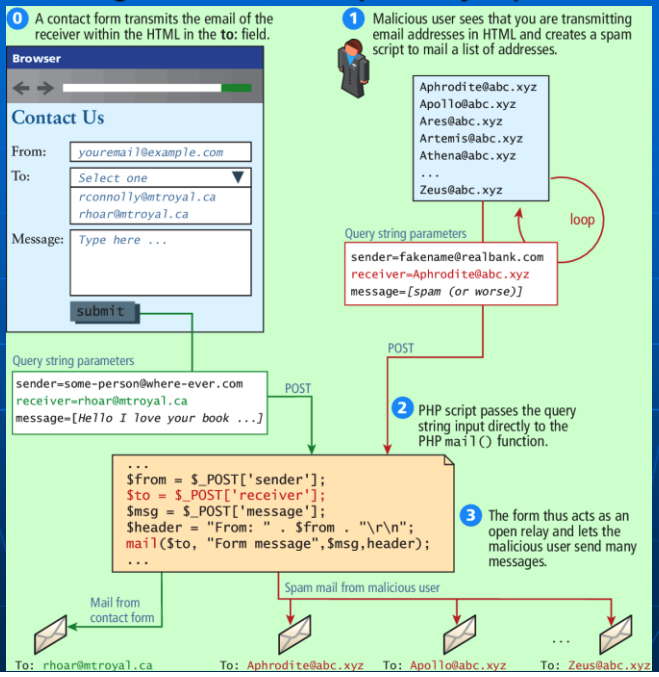
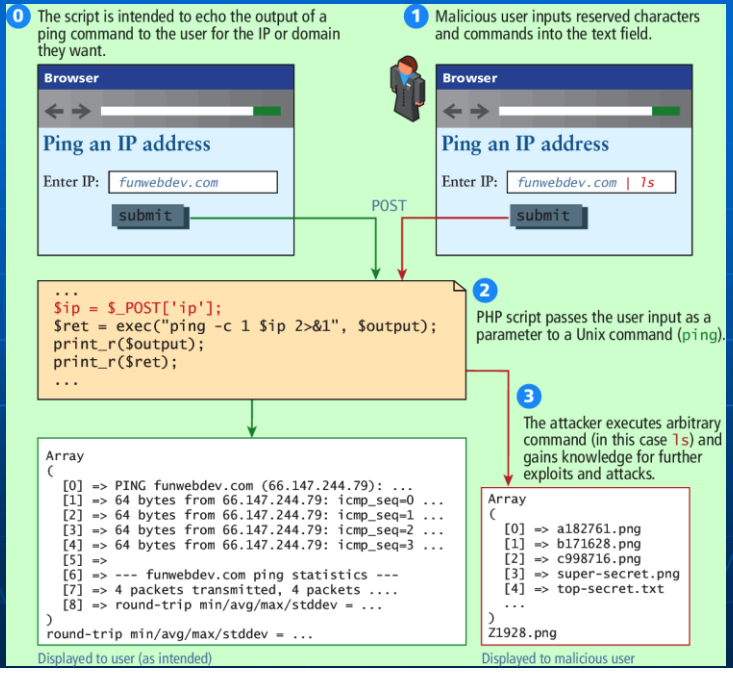


Figure 16.24 Command-line pass-through of user input



Summary and Conclusion

Q/A ?