Web Security

Mahalingam Ramkumar
Issues

- Phishing
- Spreading misinformation
- Cookies!
- Authentication
  - Domain name – DNS Security
  - Transport layer security
- Dynamic HTML
  - Java applets, ActiveX, JavaScript
- Exploiting server side security flaws
  - Server-side scripting languages
    - PHP, Java, ASP, ASP.net, CGI-bin
Cookies

- HTTP is a state-less protocol.
- How do we move from transport layer to “sessions”?  
  - Cookies
- Designed by Netscape
- Cookie fields  
  - Domain Name  
  - Path  
  - Content  
  - Expiry  
  - Secure
Cookies

- Content field – a series of name=value
- Secure field – cookie will be returned only if the server is secure (uses SSL)
- Expiry – Persistent and Non-persistent cookies
Cookies

- Cookies are created by web servers
- Stored in client machine (usually in some directory under the home directory)
- When a client (browser) connects to a server it checks for cookies for the server domain
- If a cookie is found, it sends the cookie to the server along with the connection request
Applications

- Banking
- Gaming sites
- Shopping carts
- News sites (setting user preferences)
- Web portals
- Just about every thing....
- Is it possible to do all this without cookies?
Cookie Abuse

- Tracking user habits
- Advertising agencies buy ad-space for big corporations on pages in major websites.
- Let us say an agency buys ads in N pages.
- In each page they add a link to some image for the banner ad –
  - Page 1 will have link adagency.com/image1.gif
  - Page N has link adagency.com/imageN.gif
  - (Every page has a unique link)
Profiling User Habits

- User starts with a “clean slate”
- User visits page i
- Sends http request to adagency.com for image_i.gif
- Adagency server sends back a cookie with a random but unique number
  - This will serve to distinguish the users
- Now user has a cookie identifying him!
- Every time the user goes to one of the N sites the cookie is sent to adagency.com!
Tracking User Habits

- Adagency.com gets paid for placing ads
- On top of that they “sell” collected user information!
- Cookies can be blocked
- Many websites won’t “work” though
- Fine grained control of cookies
  - Started with Mozilla
Spreading Misinformation

- Modify corporate websites
- Emulex Corporation
  - Lost 2 billion dollars due to a fake email message sent to a news agency
  - Perpetrator made a million dollars!
- Imitating web sites
- DNS Spoofing
DNS Spoofing

- Alice types in Bob's domain name
- Oscar wants to direct Alice to his page
- Oscar sends a DNS request for Bob's domain
- And also immediately sends a spoofed response
- DNS serve now has Oscar's IP (or the IP that Oscar used in the spoofed response)
- Alice receives the fake IP from the DNS server
- Don't really need to “crack” DNS servers!
DNS Spoofing

- Okay, not as simple...
- Oscar needs to fake a DNS servers IP
  - Not very difficult
- DNS requests carry a sequence number
  - Need to guess the sequence number – not so easy
- Oscar could register his own DNS server!
  - dns.oscar.com
- Alice's ISP should be made aware of oscar's dns server
  - Oscar sends a DNS request to Alice's ISP asking for the address of some machine in “oscar” domain
  - Say os1.oscar.com
DNS Spoofing

- Oscar has Alice's ISP's sequence number now
- He can send many DNS queries and then answer them too!
- Sequentially incrementing sequence number each time
- Only till some one else sends a query to Alice's ISP.
- Problem could be avoided by using a random number for each query instead of a sequence number – difficult to keep track though
Secure DNS

- DNSsec (RFC 2535)
- Every DNS zone is has a public / private key pair.
- Not possible to spoof authoritative responses
- Not all DNS servers use DNSsec
- DNSsec services
  - Proof of origin
  - Public key distribution
  - Transaction and request authentication
DNS records are grouped in resource record sets (Rrsets)

Each RRset has
- Multiple A records (for each IP the domain name translates to)
- Each record is signed

Additional records
- KEY record – contains public key of the domain
- SIG record – signed hash
DNSsec

- Signing of Rrsets done offline and inserted into DNS servers.
- No public key certificates!
- How do clients authenticate public keys?
- Could query a trusted server for verifying the public key
- Client will be armed with public keys of top-level domains
- Top-level domains will authenticate public keys of lower level domains
SSL

- Secure Socket Layer
- Purpose
  - Reliable end-to-end secure service
  - Provides a “secure TCP socket”
  - Usually used with Web browsers
    - Can be used for other applications too
  - Introduced by Netscape in 1995
  - Provided options for 40 and 128 bit keys
    - 40 bits for export
  - Submitted to IETF standards
    - Result – TLS (RFC 2246)
SSL in the Protocol Stack

<table>
<thead>
<tr>
<th>Layer</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical Layer</td>
<td>SSL Record Protocol</td>
</tr>
<tr>
<td>Data Link</td>
<td>Transport (TCP)</td>
</tr>
<tr>
<td>Network (IP)</td>
<td>HP – Handshake Protocol</td>
</tr>
<tr>
<td>Transport (TCP)</td>
<td>CSP – Change Cipher Spec</td>
</tr>
<tr>
<td>Security (SSL)</td>
<td>AP – Alert Protocol</td>
</tr>
<tr>
<td></td>
<td>HTTP</td>
</tr>
</tbody>
</table>
Broad Overview

- Perform Handshake
  - Exchange of nonces, preferences
  - Exchange of certificates, certificate chains
  - Establishment of shared key
  - Finish handshake

- Change Cipher Spec – To change ciphers on the fly

- Alert protocol for alerts
  - Unexpected message, bad record, decompression failure

- HTTP

- All performed over SSL Record Layer
Handshake

Phase 1
Security Capabilities
Session ID, Cipher Suite
Nonces

Phase 2
Server sends certificates, certificate chains, certificate request

Phase 3
Client sends certificates, (optional)

Phase 4
Change Cipher Suite
Finish handshake protocol

Client Hello
Server Hello
Certificate
Server Key Exchange
Certificate Request
Server Hello done
Certificate
Client key exchange
Certificate Verify
Change Cipher Spec
Finished
CSP
Finished
Phase 1

- Version
- Random Nonces
- Session ID
- Cipher Suite
  - Crypto algorithms supported in decreasing order of preference
  - For key exchange – RSA, Fixed Diffie Helman, Ephemeral DH, Anonymous DH, Fortezza
  - For Cipher Spec
    - Algorithms – 3DES, RC4, RC2, DES, DES40, IDEA, Fortezza
    - MAC – MD5, SHA1
    - IsExportable – True or False
- Compression Method
Phase 2,3,4

- **Server Authentication and Key Exchange**
  - Certificate, Certificate Chains
  - Server Key exchange message
  - Certificate request
  - Server done

- **Phase 3**
  - Client Authentication and Key Exchange
    - 384 bit (48 byte) pre-master key
    - If login is used, it is outside the scope of the protocol
      - Can be used safely with HTTP over SSL
SET

- Secure Electronic Transaction
- For Credit card purchases over the Internet
  - Confidentiality
  - Integrity of data
  - Authentication of card holder
  - Authentication of Merchant
  - Can be used over HTTP, SSL/TLS, IPSec
Electronic Commerce Components

Card holder

Issuer

Acquirer

Merchant

Internet
Electronic Commerce Components

Card holder
Issuer
Acquirer
Payment Gateway
Merchant
Electronic Commerce Components

Cardholder

Internet

Issuer

CA

Payment Network

Acquirer

Merchant

Payment Gateway
Participants

- Cardholder
- Merchant
- Issuer
- Acquirer
- Payment Gateway
- Certification Authority
SET Process

- Customer opens an account
  - Gets a certificate
- Merchants get certificates
  - Opens an account with an acquirer (bank)
  - Establishes a relationship with a payment gateway
- Customer verifies merchant
- Customer places order
  - Merchant verifies customer
- Merchant checks validity of card with payment gateway
- Payment gateway interacts with Acquirer
- Acquirer transfers funds from Issuer to Merchants account
Dual Signature (DS)

- Customers order has two parts
  - Order Information
  - Payment Information
- Merchant does not need to know credit card number (payment information)
- Acquirer bank does not need to know order information (order information)
Dual Signature

H – Message Digest (SHA-1)
E – RSA Encryption
KRc – Customers Private Key
POMD – Payment and Order Info MD
Purchase Request

CHC – Cardholder Certificate

Passed to Payment Gateway

Compare

CHC – Cardholder Certificate
SET Transactions

- **Purchase request**
  - Acknowledged with purchase response by the merchant

- **Payment Authorization**
  - EPI (Encrypted PI + DS + OIMD)
  - Digital Envelope (sent by customer)
  - Merchant authorization Info
    - Transaction ID signed by merchant, encrypted with one-time key
    - One time key in an envelope encrypted with gateways public key
    - CHC and Merchants certificate
    - Gateway responds with Authorization Response
    - At this point, merchant can commit to selling goods

- **Payment Capture**
  - Capture request – Initiated by merchant. Gateway performs necessary action for transfer of funds.
  - Capture response