Dos and Don’ts of Client Authentication on the Web

Kevin Fu, Emil Sit, Kendra Smith, Nick Feamster
MIT Lab for Computer Science

http://cookies.lcs.mit.edu/
cookie-eaters@mit.edu
E*TRADE Customer & Member Log On

New! Earn $50 for each new customer you refer to E*TRADE. Get started now (customer logon required)

E*TRADE User Name: [ ] Password: [ ] LOG ON [ ] Start In: [Home]

Members: Forgot your password?

Log on to OptionsLink®
(For Business Solutions clients only)

For our Chinese language investors, we now offer E*TRADE Chinese

System response and account access times may vary due to a variety of factors, including trading volumes, market conditions, system performance, and other factors.
Client authentication is solved, right?
<table>
<thead>
<tr>
<th>Site</th>
<th>Security problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSJ.com</td>
<td>crypto misuse, secret key exposed</td>
</tr>
<tr>
<td>SprintPCS.com</td>
<td>leaks authenticator in plaintext</td>
</tr>
<tr>
<td>FatBrain.com</td>
<td>predictable session ID, sequence number</td>
</tr>
<tr>
<td>PerformanceBike.com</td>
<td>predictable session ID, sequence number</td>
</tr>
<tr>
<td>highschoolalumni.com</td>
<td>circumvent password authentication</td>
</tr>
<tr>
<td>ign.com</td>
<td>circumvent password authentication</td>
</tr>
<tr>
<td>chickclick.com</td>
<td>circumvent password authentication</td>
</tr>
<tr>
<td>NEBride.com</td>
<td>circumvent password authentication</td>
</tr>
<tr>
<td>ihateshopping.net</td>
<td>circumvent password authentication</td>
</tr>
<tr>
<td>cstc.org</td>
<td>circumvent password authentication</td>
</tr>
</tbody>
</table>
## SOFTWARE GETS IT WRONG TOO

<table>
<thead>
<tr>
<th>Software product</th>
<th>Security problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allaire ColdFusion</td>
<td>session IDs, linear congruential number generator</td>
</tr>
<tr>
<td>ArsDigita ACS</td>
<td>signs ambiguous messages</td>
</tr>
<tr>
<td>Jakarta TomCat</td>
<td>session IDs, predictable random seed</td>
</tr>
</tbody>
</table>
HOW WE BROKE THESE SCHEMES

- Gathered public information
  - Usernames
  - Web server HTTP responses
  - Obtain sample authenticators

- Observe authenticators while varying parameters

- No eavesdropping
INTERROGATIVE ADVERSARY

- Treat a server as an oracle for an adaptive chosen message attack
- Adaptively query a Web server a reasonable number of times
THE INTERROGATIVE ADVERSARY DEFEATS...

• SSL client authentication? No.
• HTTP Basic or Digest authentication? No.
• Homebrew cookie authentication schemes? Maybe...
COOKIES

- A Web server can store key/value pairs on a client
- Returned in subsequent requests to the server
- Can implement login sessions
**NETSCAPE COOKIE EXAMPLE**

<table>
<thead>
<tr>
<th>Domain</th>
<th>.wsj.com</th>
</tr>
</thead>
<tbody>
<tr>
<td>Path</td>
<td>/cgi</td>
</tr>
<tr>
<td>Variable name</td>
<td>fastlogin</td>
</tr>
<tr>
<td>Value</td>
<td>bitdiddleMaRdw2J1h6Lfc</td>
</tr>
<tr>
<td>SSL?</td>
<td>FALSE</td>
</tr>
<tr>
<td>Expiration</td>
<td>941452067</td>
</tr>
</tbody>
</table>
**COOKIES FOR LOGIN SESSIONS**

1. **POST** /login.cgi

2. "Welcome in" Web page
   
   Set-Cookie: authenticator

3. GET /restricted/index.html
   
   Cookie: authenticator

4. Content of restricted page

- Enter password once per session
CASE STUDIES
The server interactive.wsj.com wishes to set a cookie that will be sent to any server in the domain .wsj.com. The name and value of the cookie are: fastlogin=

This cookie will persist until Sun Feb 25 07:26:53 2001.

Do you wish to allow the cookie to be set?
MISUSE OF CRYPTOGRAPHY: WSJ.COM

- Weaker than plaintext passwords
- Extracted secret signing key
- Can create authenticators for anyone
WSJ.COM ANALYSIS

• Design: auth = \{user, MAC_k (user)\}

• Reality: auth = 
  
  user + UNIX-crypt (user + server secret)

• Easily produce authenticator cookies

<table>
<thead>
<tr>
<th>username</th>
<th>crypt() output</th>
<th>authenticator cookie</th>
</tr>
</thead>
<tbody>
<tr>
<td>bitdiddl</td>
<td>MaRdw2J1h6Lfc</td>
<td>bitdiddlMaRdw2J1h6Lfc</td>
</tr>
<tr>
<td>bitdiddle</td>
<td>MaRdw2J1h6Lfc</td>
<td>bitdiddleMaRdw2J1h6Lfc</td>
</tr>
</tbody>
</table>
OBTAINING THE SERVER SECRET

- Adaptive chosen message attack
- Perl script queried WSJ with invalid cookies
- Runs in max of $128 \times 8$ queries rather than intended $128^8$ (1024 vs. 72057594037927936)
- 17 minutes vs. $10^9$ years
- The key is “March20”
# HOW OUR ATTACK WORKS

<table>
<thead>
<tr>
<th>Secret guess</th>
<th>username</th>
<th>crypt input</th>
<th>worked?</th>
</tr>
</thead>
<tbody>
<tr>
<td>bitdiddl</td>
<td>bitdiddl</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>bitdidd</td>
<td>bitdiddA</td>
<td>No</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>M</td>
<td>bitdidd</td>
<td>bitdiddM</td>
<td>Yes</td>
</tr>
<tr>
<td>MA</td>
<td>bitdid</td>
<td>bitdidMA</td>
<td>No</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>Ma</td>
<td>bitdid</td>
<td>bitdidMa</td>
<td>Yes</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>March20</td>
<td>b</td>
<td>bMarch20</td>
<td>Yes</td>
</tr>
</tbody>
</table>
LACK OF CRYPTOGRAPHY: HIGHSCHOOLALUMNI.COM

- Circumvent password authentication
- Cookie authenticator is the public username and public user ID
The server m27.sprintpcs.com
wishes to set a cookie that will be sent
to any server in the domain .sprintpcs.com
The name and value of the cookie are:
SPCS%5FRM=RM%5FON=Y&CN1=😊😊😊&R115=😊😊😊
This cookie will persist until Tue Mar 27 19:01:45 2001

Do you wish to allow the cookie to be set?
LEAKING SECRETS: SPRINTPCS.COM

- Secure content can leak through plaintext channels
- A cookie has flag to require SSL
- User logs in with HTTPS, then clicks back to main HTTP page
- Vulnerable to passive eavesdropper
HINTS FOR CLIENT AUTHENTICATION

- Limit the lifetime of authenticators
- Make authenticators unforgeable
- Sign what you mean
LIMIT THE LIFETIME OF AUTHENTICATORS

- Browsers cannot be trusted to expire cookies
- No revocation of WSJ cookies
MAKE AUTHENTICATORS UNFORGEABLE

- Prevent modification of the cookie
- Do not allow bypass of password authentication
- Highschoolalumni.com
SIGN WHAT YOU MEAN!

- badauth = sign (username + expiration, key)
  - (Alice, 21-Apr-2001) → sign (Alice21-Apr-2001, key)
  - (Alice2, 1-Apr-2001) → sign (Alice21-Apr-2001, key)

- Same authenticator!

- Use unambiguous representation or delimiters
A SCHEME THAT WORKS

\[ \text{auth} = \text{expire} + \text{data} + \text{MAC}_k(\text{expire} + \text{data}) \]

where \text{MAC} could be HMAC-SHA1, 
\text{data} could be a username or capability, and 
‘+’ denotes concatenation with a delimiter

Secure against \textit{interrogative} adversary
SUMMARY

• Many schemes *easily* broken
• Following hints can prevent vulnerabilities
• Juicy details in our technical report
• Cookies are limited; live with it or move on
JOIN US

DONATE YOUR COOKIES FOR ANALYSIS*

http://cookies.lcs.mit.edu/
cookie-eaters@mit.edu

*may be tax deductible
All your cookie are belong to us