The Failure of Client Authentication on the Web

Kevin Fu, Emil Sit, Kendra Smith, Nick Feamster

MIT Lincoln Laboratory April 18, 2001
There are fine lines.

• Don't do this.

• There are fine lines.

Caveat haxor
The Web Protocol: HTTP

Web browser

[GET /index.html]

Web server

[Contents of index.html]
What is authentication?
Why is client authentication on the Web difficult?

- Limited interface.
- Hard-to-manage client-side storage.
- Solutions that exist are not deployable (e.g., personal certificates)
Case studies of Web authentication

- Misuse of cryptography: WSJ.com
- SSL and plain HTTP do not work together:
  - SprintPCS

- Predictable sequence numbers: Fastbrain.com
- Security through obscurity:
  - HighSchoolAlumni.com

- Letting clients name the price: InstanShop

- SprintPCS

SSL and plain HTTP do not work together:

High School Alumni.com
Cookies: What are they?

- A server can store key/value pairs on a client.
- The client sends previously set cookies to the server.
The Web protocol with cookies

GET /restricted/index.html

Cookie: authenticator

Set-Cookie: authenticator

"Welcome in Web page"

POST /login.cgi

Web browser

Web server

Content of restricted page
Netscape cookie example

wsj.com FALSE /cgi FALSE SSL? Path FALSE Expiration Variable 
941452067 fastlogin name fastlogin MaRdw2J1h6Lfc
Taxonomy of Adversaries

- Oracle: Can query a service.
- Passive: Can listen to network traffic.
- Active: Can listen, modify, and insert network traffic.
SSL and plain HTTP do not work together.

SprintPCS.com

Problem: Secure content can leak through plaintext channels.

Cookie file has flag to require SSL.

User logs in with HTTPS, then clicks back to main HTTP page.

Vulnerable to eavesdroppers.
Do you wish to allow the cookie to be set?

This cookie will persist until the Mar 27 19:01:45 2001

The name and value of the cookie are:

SPCS%5FRETRY=RM%5FRETRY=Y&CSN=38&amp1115=

The server at sprintpcs.com wishes to set a cookie that will be sent

Information entered is:

Customer Sign In

Password:

MongoDB

This information will be saved for the next time you sign in.
Letting clients name the price: Instant Shop

Problem: Trusting clients not to modify HTML variables.

Price determined by hidden variable in Web page.

Make a local copy of the Web page. Modify it.
Instant Shop example: What's inside

```
<html>
<body>
<form action="commit-sale.cgi">
<input type="hidden" name="item1" value="10">
Batteries $10
<br>
<input type="hidden" name="item2" value="99">
Biology textbook $99
<br>
<input type="hidden" name="item3" value="25">
Britney Spears CD $25
<br>
<input type="submit" value="Confirm purchase">
</form>
</body>
</html>
```
Instant Shop Example: Malicious Client

<body>
<form action='commit-sale.cgi'
 input type='submit' Confirn purchase>
<input type='hidden' name='item1' value='0'> Batteries $10
<br>
<input type='hidden' name='item2' value='0'> Biology textbook $99
<br>
<input type='hidden' name='item3' value='0'> Britney Spears CD $25
<br>
<form action='commit-sale.cgi'
 input type='submit' CONFIRM Purchase>
</body>
</html>
Problem: No cryptographic authentication at all. Cookie authenticator is the public username and public user ID.

Security through obscurity:

HighSchoolAlumni.com
Problem: Customer can determine the authenticator for any other user.

Authenticators are sequence numbers in the URL.

Guess a victim's sequence number by decrementing.

Access to personal information, receive password by email.

Predictable sequence numbers: fatbrain.com
Thanks and we hope you enjoy the flexibility available with your account.

For detailed information on what you can do with your account, click the "More..." link next to your topic of interest or simply scroll down the page.

Your Account

Welcome to your Account.
“It’s frustrating that programmers continue to fall prey to the same old tricks. Simple problems like lazy sequence numbers and buffer overflows in most cases can be easily eliminated if we were a little vigilant about sound design and solid code reviews. I just love being at work on a Friday at midnight managing unscheduled production releases. :)”

—Chris Grant

Fatbrain Response
Can purchase articles. Optional stock portfolio tracking.

Half million paid-subscriber accounts. Stateless servers.

Wanted to authenticate paid subscribers with

wsj.com
Misuse of Cryptography: WSJ.com

Problem: Cryptography used incorrectly can be worse than no cryptography at all.

Easily guessable authenticator.

Given a username, our Perl script produces the authenticator.
WSJ.com analysis: the crypt() hash function

- Produces a hash.
- Ignores all input after the 8th character.
- Takes an 8-character input and salt.
The fastlogin cookie lasts forever

Lack of revocation.

Using your fastlogin cookie to produce another:

user + cryptr (user + rotating server secret) = fastlogin

Wsj.com analysis continued
The key is “March20.”

1 sec/query yields 17 minutes vs. 109 years.

1288 (1024 vs. 72057594037927936).

Runs in max 128 × 8 queries rather than intended.

Perl script queried WSJ with invalid cookies.

Programming.

Adaptive chosen plaintext attack (dynamic).

How did we obtain the rotating server secret?
How our attack works

Pad guess username crypt input worked?

bit didl bit didl Yes
...

bit didl bit didl bit  March 20  b  March 20
...

bit didl bit didl Yes
...

bit didl bit didl MA
...

bit didl bit didl No
...

bit didl bit didl MA
...

bit didl bit didl M
...

bit didl bit didl No
...

bit didl bit didl A

Pad guess username crypt input worked?
“...about the factors affecting design decisions, it is certainly result of time to market considerations. We simply didn’t have clear security requirements defined within the group and outside the group. So, we did what worked. We fixed a better encryption algorithm, but hit a bug that we couldn’t fix, so we implemented another one that worked even though the architect in charge was fully aware of its short-comings. You must understand that I’m giving my read on the situation since I’ve joined WSJ.com just 5 weeks ago."

— Javeh Saleh

Vice President, Technology
Interactive Business Technology Services, WSJ.com
Why do sites use cookies for authentication?

• Popular browsers implement cookies.
• HTTP digest authentication is not deployed.
• HTTP authentication exposes passwords in cleartext.
• SSL is computationally expensive.

SSL is computationally expensive.
Simpleschemes that work

Active. Same as passive, but over SSL.

where MAC could be HMAC-SHA1

-passive. cookie = \exp \cdot \text{MAC}_{\text{URL}} + dx \cdot \exp \cdot \text{MAC}_{\text{URL}}^d

• Oracle. cookie = \text{username} + \text{password}

Simple schemes that work
Server authentication is difficult too.

• Netscape demo.

• Hostname.

• Caching SSL sessions on IP addresses rather than

Server authentication is difficult too.
Conclusions

Keep It Simple, Stupid (KISS).

Subtle assumptions can lead to insecurity.

No company wants to be the first to publish a cookie authentication scheme.

Work to appear on http://cookies.ics.mit.edu/

and USENIX Sec01.
If you leave the door open...
What is SSL: channel security

- Integrity protection
- Authentication
- Confidentiality
Certificates contain a public key, metadata, and a signature by a trusted third party.
• Certificate Authorities (CAs)

  Example: VeriSign.

  Trusted third party with well-known public key.

  Certifies who belongs to a public key.
What does a CA-issued certificate mean?

No one knows exactly.

That a public key belongs to someone who has lots of paper trails associated to a company related to a hostname.

That a public key belongs to someone who is associated in some way with a hostname.

That a public key belongs to someone who is to represent a hostname.

That a public key belongs to someone authorized.
How to get a Verisign certificate

- Pay Verisign ($300)
- City of Cambridge License ($20)
- Letterhead from company ($0)
- Notarized document (need driver's license) ($0)
Neither Netscape nor Microsoft have published their rule set for deciding which CA roots to include in browsers.

Every CA is equally trusted.

A single bad CA can disrupt authentication for the whole system.

SSL pitfalls: Default CAs in browsers.
These certificates identify the certificate issuers that you accept:
SSL pitfalls: CA revocation

[Sun CA]

- What if a CA itself is compromised?
- No way to revoke a certificate.
- Certificates last for a long time, typically a year.

SSL pitfalls: CA revocation
Insecure.

Because of an insecure implementation, SSL was insecure.

- Predicted sessions keys.
- Two Berkeley graduate students were able to predict session keys.
- Netscape used predictable numbers to generate SSL session keys.

SSL pitfalls: Random number generation
SSL pitfalls: End-to-end content

SSL authentication

SSL authenticates servers, not content. [Akamai]

SSL pitfalls: End-to-end content

SSL authentication
SSL Pitfalls: Perfect Forward Secrecy

- Compromised server private key decrypts future and past traffic.

SSL Pitfalls: Perfect Forward Secrecy