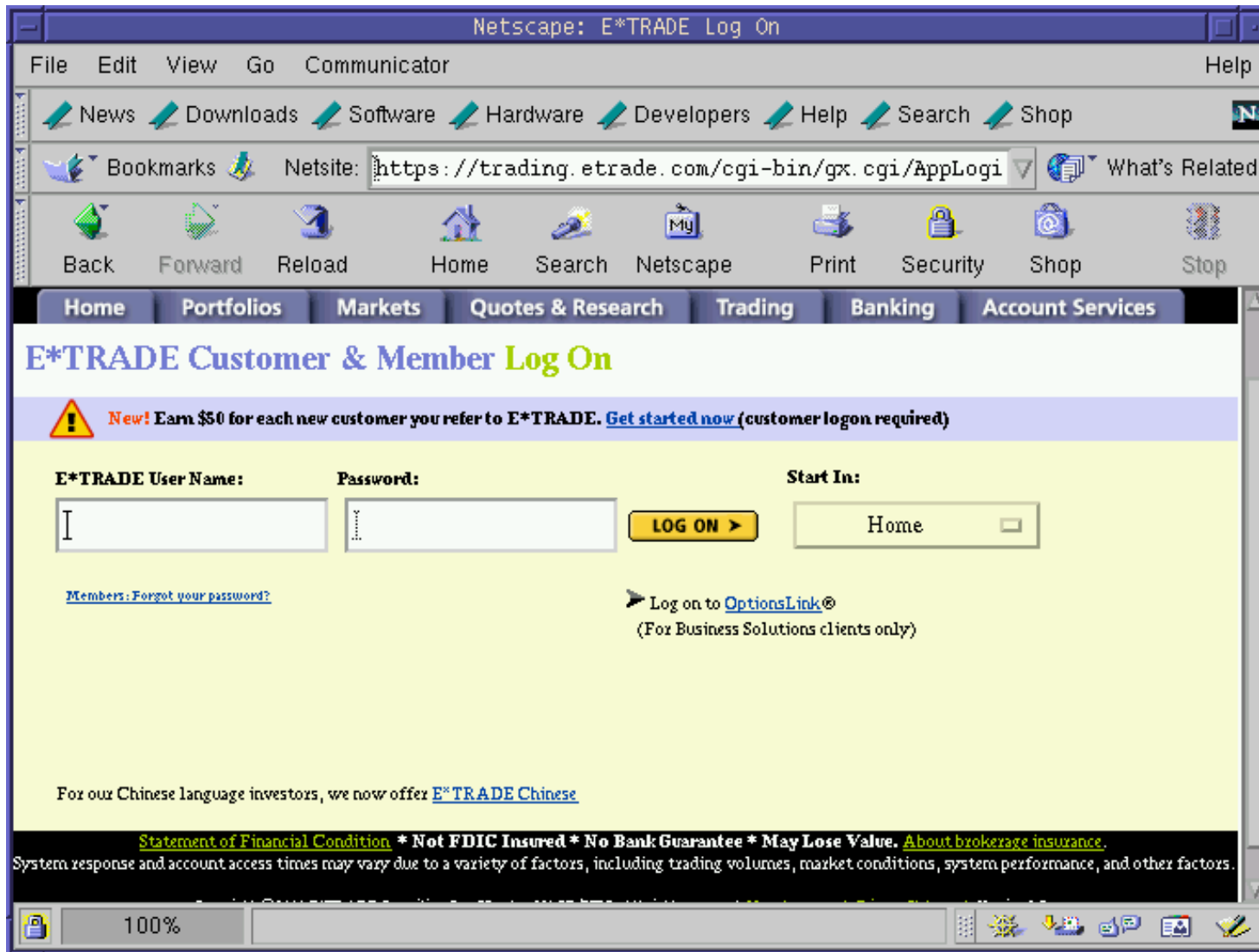


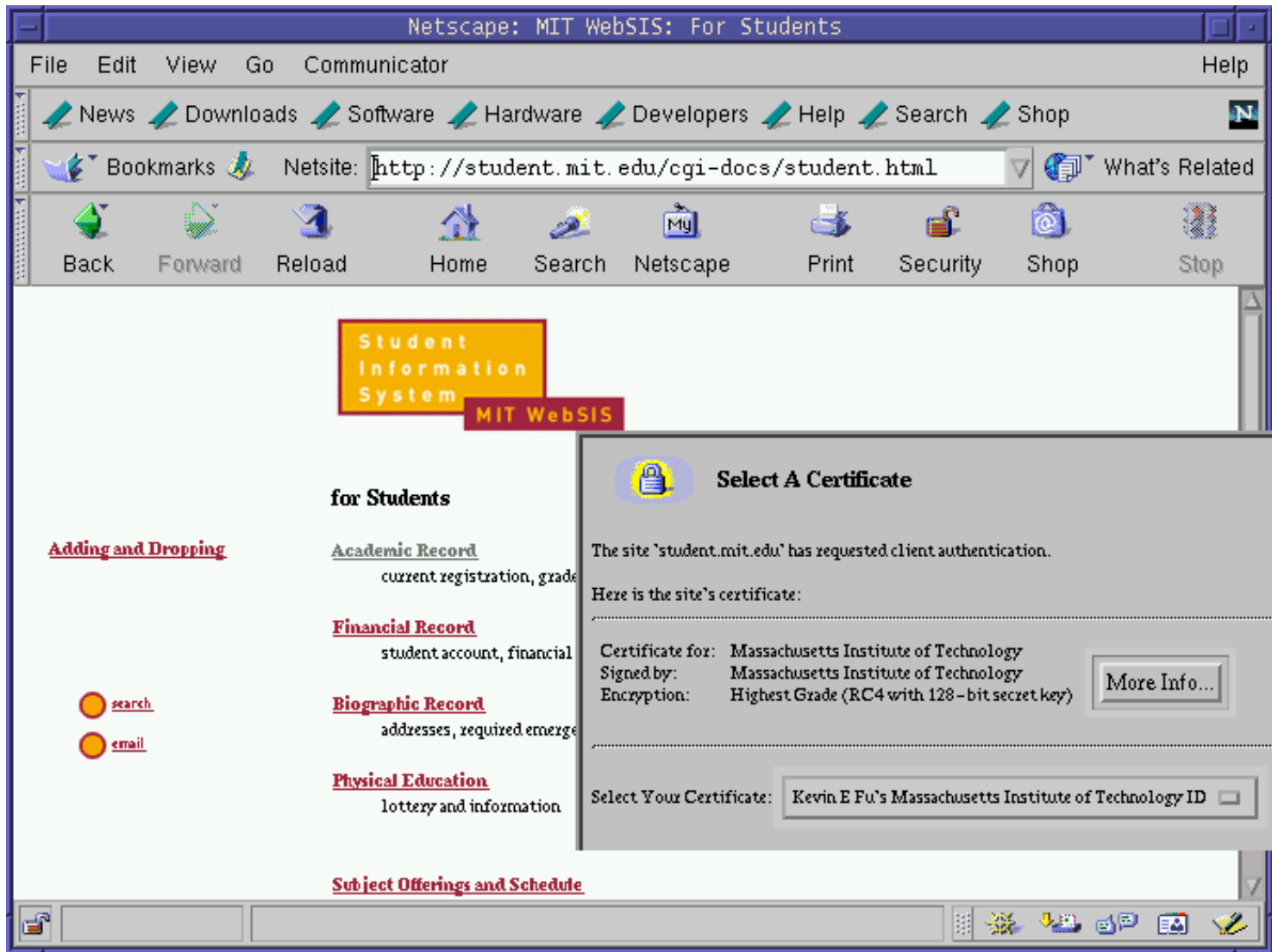
Dos and Don'ts of Client Authentication on the Web

Kevin Fu

**UMass-Amherst
Department of Computer Science
www.cs.umass.edu**

Based on USENIX Security 2001 paper by same name.
Versions of this talk were given several times. History on:
<http://www.cs.umass.edu/~kevinfu/talks.html>





What this talk is about

- Improving the security of client authentication on the Web

Where are we now?

- We have HTTP authentication



The image shows a standard Windows authentication dialog box. The title bar reads "Connect to snafu.lcs.mit.edu" and includes help and close buttons. The main area has a blue header with a key icon and the text "Password Required". Below this, there are two input fields: "User name:" with a dropdown menu and "Password:" with a text box. A checkbox labeled "Remember my password" is located below the password field. At the bottom, there are "OK" and "Cancel" buttons.

Connect to snafu.lcs.mit.edu

Password Required

User name:

Password:

Remember my password

OK Cancel

Where are we now?

- We have HTTP authentication
- We've had SSL for nearly a decade

Where are we now?

- We have HTTP authentication
- We've had SSL for nearly a decade
- Client authentication should be **easy**, right?

Many Web sites get it wrong

Site	Security problem
WSJ.com	crypto misuse, secret key exposed
tiffany.com	SQL injection
opentable.com	guessable user IDs
cooking.com	guessable user IDs
SprintPCS.com	leaks authenticator in plaintext
FatBrain.com	predictable session ID
HighSchoolAlumni.com	circumvent password authentication
PerformanceBike.com	predictable session ID
ihateshopping.net	circumvent password authentication

Toolkits are vulnerable too

Toolkit	Security problem
BlueMartini	missing authentication check
Allaire ColdFusion	predictable session IDs, LCNG
ArsDigita ACS	signs ambiguous messages
Jakarta TomCat	predictable session IDs, random seed
PHP	session IDs based on time of day

How is it done?

So how do Web sites implement
user authentication?

Cookies: what are they?

- A Web server can store key/value pairs on a client
- The browser resends cookies in subsequent requests to the server
- Cookies can implement login sessions

Sample cookie

domain	.wsj.com
Path	/cgi
SSL?	FALSE
Expiration	941452067
Variable name	fastlogin
Value	bitdiddleMaRdw2J1h6Lfc

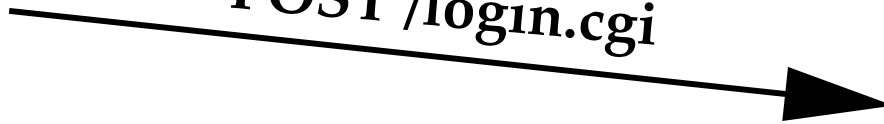
Cookies for login sessions

Web browser

Web server

1

POST /login.cgi



Cookies for login sessions

Web browser

Web server

1

POST /login.cgi

2

*"Welcome in" Web page
Set-Cookie: authenticator*



Cookies for login sessions

Web browser

Web server

1

POST /login.cgi



2

"Welcome in" Web page
Set-Cookie: authenticator



3

GET /restricted/index.html
Cookie: authenticator



Cookies for login sessions

Web browser

Web server


1 *POST /login.cgi*



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



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



4 *Content of restricted page*



What adversaries do we fear?

Active adversary

Passive adversary

Interrogative adversary

- Adaptively query a server
- Eavesdrop on traffic
- Modify/inject traffic, man-in-the-middle attack

A system must **AT LEAST** protect against the interrogative adversary!

Interrogative adversary

- Adaptively query a Web server a reasonable number of times
- Treat server as an oracle for an adaptive chosen message attack
- Extremely limited, but surprisingly powerful

Types of breaks

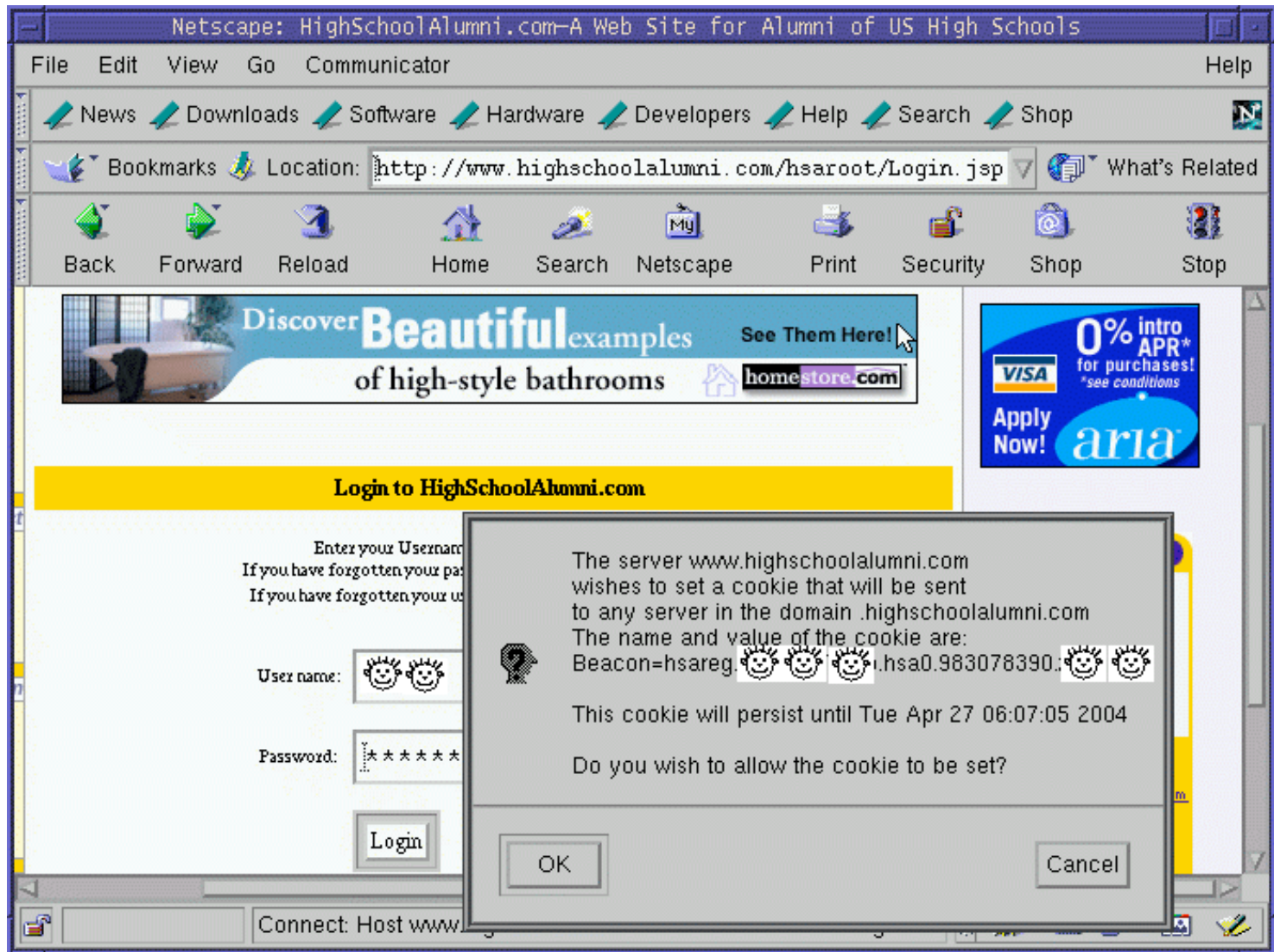
- Replay
- Existential forgery
- Selective forgery
- Total break

The cookie crumbles...

Many Web sites that have invented their own homebrew cookie-based authentication schemes.

Case studies of Web authentication

- Lack of cryptography:
HighSchoolAlumni.com
- Trusting user input: Instant Shop
- Leaking secrets: SprintPCS.com
- Predictable sequence numbers: FatBrain.com
- Missing authentication check: BlueMartini
- Misuse of cryptography: WSJ.com





The server www.highschoolalumni.com
wishes to set a cookie that will be sent
to any server in the domain .highschoolalumni.com

The name and value of the cookie are:

Beacon=hsareg. hsa0.983078390.xWJjw4

This cookie will persist until Tue Apr 27 06:07:05 2004

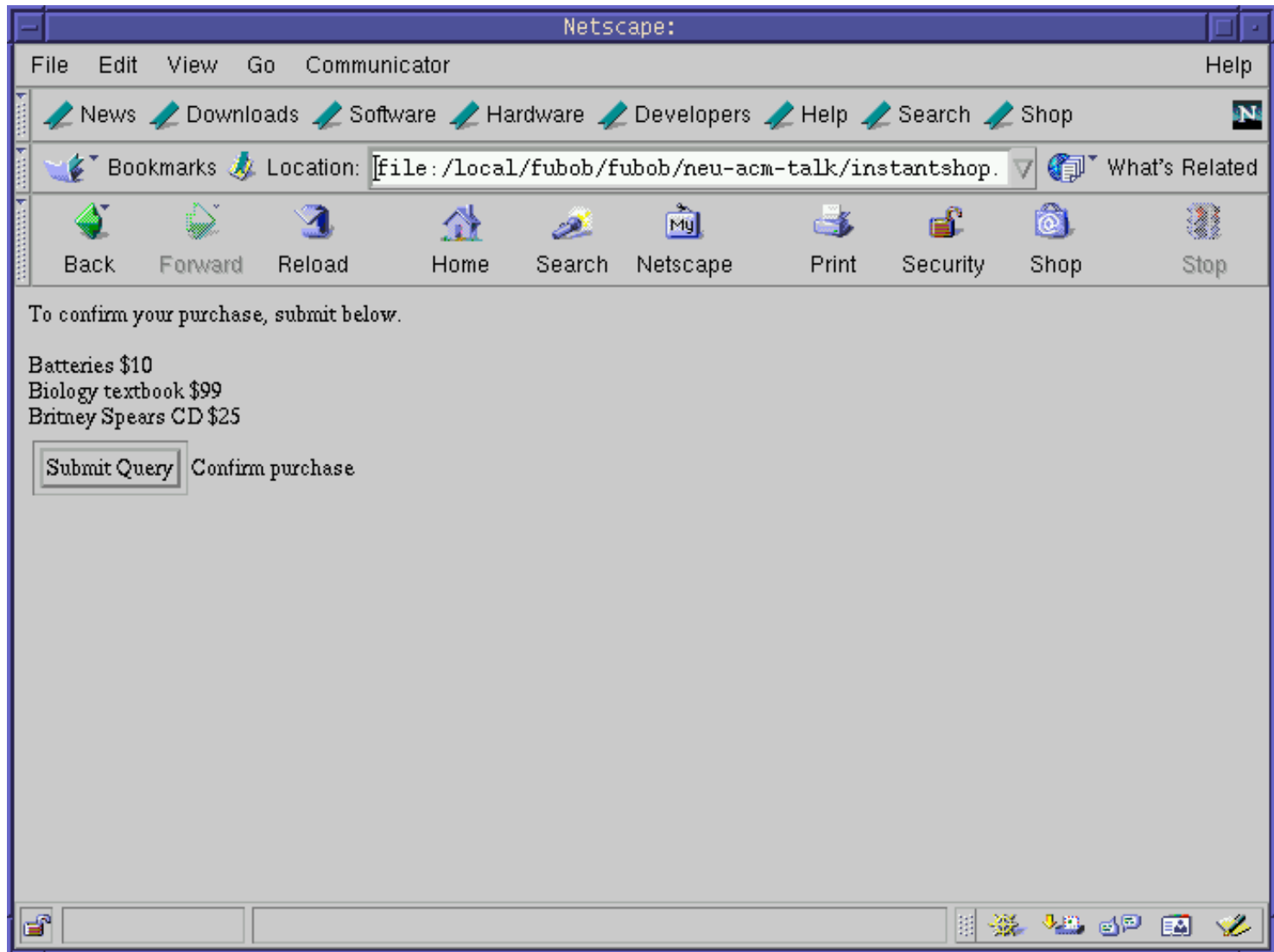
Do you wish to allow the cookie to be set?

OK

Cancel

Lack of cryptography

- Site: HighSchoolAlumni.com
- Problem: No cryptographic authentication
- Adversary: Interrogative
- Break: Universal forgery
- Today: Sold to another reunion site



Instant Shop: What's inside

```
<form action=commit_sale.cgi>
```

```
<input type=hidden name=item1 value=10>Batteries  
$10
```

```
<input type=hidden name=item2 value=99>Biology  
textbook $99
```

```
<input type=hidden name=item3 value=25>Britney  
Spears CD $25
```

```
<input type=submit>Confirm purchase
```

```
</form>
```

Instant Shop: Malicious user

```
<form action=commit_sale.cgi>
```

```
<input type=hidden name=item1 value=0>Batteries  
$10
```

```
<input type=hidden name=item2 value=0>Biology  
textbook $99
```

```
<input type=hidden name=item3 value=0>Britney  
Spears CD $25
```

```
<input type=submit>Confirm purchase
```

```
</form>
```

Trusting user input

- Site: Instant Shop
- Problem: Server trusts users not to modify HTML variables
- Adversary: Interrogative
- Today: Out of business

Netscape: Sprint PCS - Your Account Manager

File Edit View Go Communicator Help

News Downloads Software Hardware Developers Help Search Shop

Bookmarks Location: https://m27.sprintpcs.com/manage/general_manage_login.asp What's Related

Back Forward Reload Home Search Netscape Print Security Shop Stop

Sprint **Sprint PCS®**

Shop Manage

My Account My Services Customer Care Tutorials ? Help

Manage Your Sprint PCS Account Online

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?

Cancel

Customer Sign In

Enter Your Sprint PCS Phone Number

617-

Enter Your Account Password



Remember me

Sign In

[Get my Password](#)

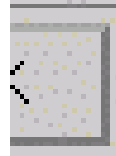
Connect: Host m27.sprintpcs.com contacted. Waiting for reply...

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?



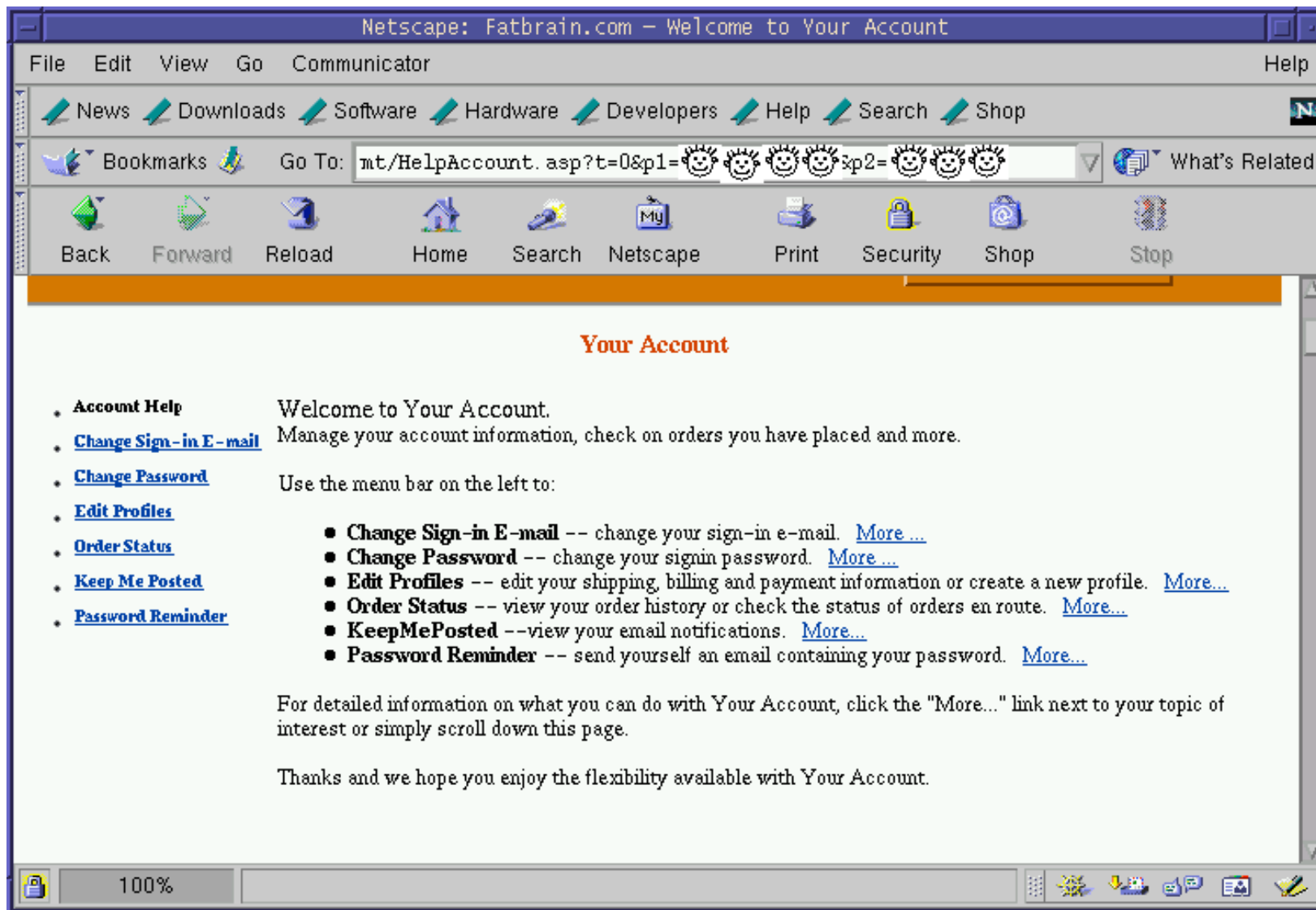
Cancel



Connect: Host m27.sprintpcs.com contacted. Waitir

Leaking secrets

- Site: SprintPCS.com
- Problem: Secure content can leak through plaintext channels
- Adversary: Eavesdropper
- Break: Replay
- Today: A leading provider of mobile phone service...

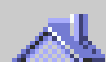


Netscape: Fatbrain.com - Welcome to Your Account

Communicator

Software  Hardware  Developers  Help  Search  Shop

0: mt/HelpAccount.asp?t=0&p1=fubob@mit.edu&p2=: 



FatBrain URL authenticator

Start: [https://www.fatbrain.com/HelpAccount.asp?](https://www.fatbrain.com/HelpAccount.asp?t=0&p1=attacker@mit.edu&p2=540555758)

[t=0&p1=attacker@mit.edu&p2=540555758](https://www.fatbrain.com/HelpAccount.asp?t=0&p1=attacker@mit.edu&p2=540555758)

Try: [https://www.fatbrain.com/HelpAccount.asp?](https://www.fatbrain.com/HelpAccount.asp?t=0&p1=victim@mit.edu&p2=540555757) ✘

[t=0&p1=victim@mit.edu&p2=540555757](https://www.fatbrain.com/HelpAccount.asp?t=0&p1=victim@mit.edu&p2=540555757)

Target: [https://www.fatbrain.com/HelpAccount.asp?](https://www.fatbrain.com/HelpAccount.asp?t=0&p1=victim@mit.edu&p2=540555752)

[t=0&p1=victim@mit.edu&p2=540555752](https://www.fatbrain.com/HelpAccount.asp?t=0&p1=victim@mit.edu&p2=540555752)

FatBrain URL authenticator

Start: [https://www.fatbrain.com/HelpAccount.asp?](https://www.fatbrain.com/HelpAccount.asp?t=0&p1=attacker@mit.edu&p2=540555758)

[t=0&p1=attacker@mit.edu&p2=540555758](https://www.fatbrain.com/HelpAccount.asp?t=0&p1=attacker@mit.edu&p2=540555758)

Try: [https://www.fatbrain.com/HelpAccount.asp?](https://www.fatbrain.com/HelpAccount.asp?t=0&p1=victim@mit.edu&p2=540555756) ✘

[t=0&p1=victim@mit.edu&p2=540555756](https://www.fatbrain.com/HelpAccount.asp?t=0&p1=victim@mit.edu&p2=540555756)

Target: [https://www.fatbrain.com/HelpAccount.asp?](https://www.fatbrain.com/HelpAccount.asp?t=0&p1=victim@mit.edu&p2=540555752)

[t=0&p1=victim@mit.edu&p2=540555752](https://www.fatbrain.com/HelpAccount.asp?t=0&p1=victim@mit.edu&p2=540555752)

FatBrain URL authenticator

Start: [https://www.fatbrain.com/HelpAccount.asp?](https://www.fatbrain.com/HelpAccount.asp?t=0&p1=attacker@mit.edu&p2=540555758)

[t=0&p1=attacker@mit.edu&p2=540555758](https://www.fatbrain.com/HelpAccount.asp?t=0&p1=attacker@mit.edu&p2=540555758)

Try: [https://www.fatbrain.com/HelpAccount.asp?](https://www.fatbrain.com/HelpAccount.asp?t=0&p1=victim@mit.edu&p2=540555755) ✘

[t=0&p1=victim@mit.edu&p2=540555755](https://www.fatbrain.com/HelpAccount.asp?t=0&p1=victim@mit.edu&p2=540555755)

Target: [https://www.fatbrain.com/HelpAccount.asp?](https://www.fatbrain.com/HelpAccount.asp?t=0&p1=victim@mit.edu&p2=540555752)

[t=0&p1=victim@mit.edu&p2=540555752](https://www.fatbrain.com/HelpAccount.asp?t=0&p1=victim@mit.edu&p2=540555752)

FatBrain URL authenticator

Start: [https://www.fatbrain.com/HelpAccount.asp?
t=0&p1=attacker@mit.edu&p2=540555758](https://www.fatbrain.com/HelpAccount.asp?t=0&p1=attacker@mit.edu&p2=540555758)

Try: [https://www.fatbrain.com/HelpAccount.asp? ✘
t=0&p1=victim@mit.edu&p2=540555754](https://www.fatbrain.com/HelpAccount.asp?t=0&p1=victim@mit.edu&p2=540555754)

Target: [https://www.fatbrain.com/HelpAccount.asp?
t=0&p1=victim@mit.edu&p2=540555752](https://www.fatbrain.com/HelpAccount.asp?t=0&p1=victim@mit.edu&p2=540555752)

FatBrain URL authenticator

Start: [https://www.fatbrain.com/HelpAccount.asp?
t=0&p1=attacker@mit.edu&p2=540555758](https://www.fatbrain.com/HelpAccount.asp?t=0&p1=attacker@mit.edu&p2=540555758)

Try: [https://www.fatbrain.com/HelpAccount.asp? ✘
t=0&p1=victim@mit.edu&p2=540555753](https://www.fatbrain.com/HelpAccount.asp?t=0&p1=victim@mit.edu&p2=540555753)

Target: [https://www.fatbrain.com/HelpAccount.asp?
t=0&p1=victim@mit.edu&p2=540555752](https://www.fatbrain.com/HelpAccount.asp?t=0&p1=victim@mit.edu&p2=540555752)

FatBrain URL authenticator

Start: [https://www.fatbrain.com/HelpAccount.asp?](https://www.fatbrain.com/HelpAccount.asp?t=0&p1=attacker@mit.edu&p2=540555758)

[t=0&p1=attacker@mit.edu&p2=540555758](https://www.fatbrain.com/HelpAccount.asp?t=0&p1=attacker@mit.edu&p2=540555758)

Try: [https://www.fatbrain.com/HelpAccount.asp?](https://www.fatbrain.com/HelpAccount.asp?t=0&p1=victim@mit.edu&p2=540555752) ✓

[t=0&p1=victim@mit.edu&p2=540555752](https://www.fatbrain.com/HelpAccount.asp?t=0&p1=victim@mit.edu&p2=540555752)

Target: [https://www.fatbrain.com/HelpAccount.asp?](https://www.fatbrain.com/HelpAccount.asp?t=0&p1=victim@mit.edu&p2=540555752)

[t=0&p1=victim@mit.edu&p2=540555752](https://www.fatbrain.com/HelpAccount.asp?t=0&p1=victim@mit.edu&p2=540555752)

Predictable sequence numbers

- Site: FatBrain.com
- Problem: Customer can determine the authenticator for any other user
- Adversary: Interrogative
- Break: Selective forgery
- Today: Acquired by Barnes & Noble

FatBrain response

“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 as programmers would be 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. :)”

Missing authentication check

- Sites: saksfifthavenue.com, kohls.com, iomega.com, et al
- Problem: Customers can download order history of all users
- Adversary: Interrogative
- Break: Universal forgery
- Today: The sites have added the check

BlueMartini: missing authentication check

https://www.saksfifthavenue.com/

POST /myaccount/order_history_new.jsp HTTP/1.0

Host: www.saksfifthavenue.com

bmForm=order_history_new&

bmHidden=VIEW_ORDER<>&

VIEW_ORDER<>orh_id=12366456

Netscape: WSJ.com Home Page

File Edit View Go Communicator Help

News Downloads Software Hardware Developers Help Search Shop

Bookmarks Netsite: What's Related

Back Forward Reload Home Search Netscape Print Security Shop Stop

WSJ.com THE WALL STREET JOURNAL.

Other Views:
[ASIA](#) [EUROPE](#)
[Set Default View](#)

U.S. View

Free U.S. Quotes
Enter Symbol Here

WSJ.com Subscribers
Go Directly To:

Or [LOG IN](#)

WSJ.COM SUBSCRIBERS ONLY

Top Business News

- [Davis Says California Has Deal With Utility](#)
- [Employers Plan Slight Scaling Back](#)

100% 100% of 7K (at 227 bytes/se)

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= [REDACTED]

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

Do you wish to allow the cookie to be set?

WSJ.com login process

- User enters name and password
- If the password is correct, WSJ.com issues a cookie
- User surfs to restricted content and attaches cookie
- If the cookie is authentic, WSJ.com returns content

WSJ.com analysis

- Design: $\text{cookie} = \{\text{user}, \text{MAC}_k(\text{user})\}$
- Reality: $\text{cookie} =$
 $\text{user} + \text{UNIX-crypt}(\text{user} + \text{server secret})$

WSJ.com analysis cont.

username	crypt() Output	Authenticator cookie
bitdiddl	MaRdw2J1h6Lfc	bitdiddlMaRdw2J1h6Lfc
bitdiddle	MaRdw2J1h6Lfc	bitdiddleMaRdw2J1h6Lfc

- Usernames matching first 8 characters have same authenticator
- No expiration

Obtaining the server secret?

- Adaptive chosen message attack
- Perl script queried WSJ with invalid cookies
- Runs in max 128×8 queries rather than intended 128^8 (1024 vs. 72057594037927936)
- 1 sec/query yields 17 minutes vs. 10^9 years
- The key is “March20”

How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	

How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	✓
A	bitdidd	bitdiddA	✗

How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	✓
B	bitdidd	bitdiddB	✗

How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	✓
C	bitdidd	bitdiddC	✗

How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	✓
D	bitdidd	bitdiddD	✗

How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	✓
E	bitdidd	bitdiddE	✗

How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	✓
F	bitdidd	bitdiddF	✗

How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	✓
G	bitdidd	bitdiddG	✗

How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	✓
H	bitdidd	bitdiddH	✗

How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	✓
I	bitdidd	bitdiddI	✗

How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	✓
J	bitdidd	bitdiddJ	✗

How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	✓
K	bitdidd	bitdiddK	✗

How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	✓
L	bitdidd	bitdiddL	✗

How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	
M	bitdidd	bitdiddM	

How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	✓
M	bitdidd	bitdiddM	✓
MA	bitdid	bitdidMA	✗

How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	✓
M	bitdidd	bitdiddM	✓
MB	bitdid	bitdidMB	✗

How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	✓
M	bitdidd	bitdiddM	✓
MC	bitdid	bitdidMC	✗

How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	✓
M	bitdidd	bitdiddM	✓
MD	bitdid	bitdidMD	✗

How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	✓
M	bitdidd	bitdiddM	✓
ME	bitdid	bitdidME	✗

How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	✓
M	bitdidd	bitdiddM	✓
MF	bitdid	bitdidMF	✗

How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	✓
M	bitdidd	bitdiddM	✓
MG	bitdid	bitdidMG	✗

How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	✓
M	bitdidd	bitdiddM	✓
MH	bitdid	bitdidMH	✗

How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	✓
M	bitdidd	bitdiddM	✓
MI	bitdid	bitdidMI	✗

How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	✓
M	bitdidd	bitdiddM	✓
MJ	bitdid	bitdidMJ	✗




How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	✓
M	bitdidd	bitdiddM	✓
MK	bitdid	bitdidMK	✗

How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	✓
M	bitdidd	bitdiddM	✓
ML	bitdid	bitdidML	✗

How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	
M	bitdidd	bitdiddM	
Ma	bitdid	bitdidMa	

How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	✓
M	bitdidd	bitdiddM	✓
Ma	bitdid	bitdidMa	✓
MaA	bitdi	bitdiMaA	✗

How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	
M	bitdidd	bitdiddM	
Ma	bitdid	bitdidMa	
Mar	bitdi	bitdiMar	

How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	✓
M	bitdidd	bitdiddM	✓
Ma	bitdid	bitdidMa	✓
Mar	bitdi	bitdiMar	✓
Marb	bitd	bitdMarb	✗

How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	✓
M	bitdidd	bitdiddM	✓
Ma	bitdid	bitdidMa	✓
Mar	bitdi	bitdiMar	✓
Marc	bitd	bitdMarc	✓

How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	✓
M	bitdidd	bitdiddM	✓
Ma	bitdid	bitdidMa	✓
Mar	bitdi	bitdiMar	✓
Marc	bitd	bitdMarc	✓
Marcg	bit	bitMarcg	✗

How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	✓
M	bitdidd	bitdiddM	✓
Ma	bitdid	bitdidMa	✓
Mar	bitdi	bitdiMar	✓
Marc	bitd	bitdMarc	✓
March	bit	bitMarch	✓

How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	✓
M	bitdidd	bitdiddM	✓
Ma	bitdid	bitdidMa	✓
Mar	bitdi	bitdiMar	✓
Marc	bitd	bitdMarc	✓
March	bit	bitMarch	✓
March1	bi	biMarch1	✗

How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	✓
M	bitdidd	bitdiddM	✓
Ma	bitdid	bitdidMa	✓
Mar	bitdi	bitdiMar	✓
Marc	bitd	bitdMarc	✓
March	bit	bitMarch	✓
March2	bi	biMarch2	✓

How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	✓
M	bitdidd	bitdiddM	✓
Ma	bitdid	bitdidMa	✓
Mar	bitdi	bitdiMar	✓
Marc	bitd	bitdMarc	✓
March	bit	bitMarch	✓
March2	bi	biMarch2	✓
March2/	b	bMarch2/	✗

How our attack works

Secret guess	username	crypt input	worked?
	bitdiddl	bitdiddl	✓
M	bitdidd	bitdiddM	✓
Ma	bitdid	bitdidMa	✓
Mar	bitdi	bitdiMar	✓
Marc	bitd	bitdMarc	✓
March	bit	bitMarch	✓
March2	bi	biMarch2	✓
March20	b	bMarch20	✓

Misuse of cryptography

- Site: WSJ.com
- Problem: Weaker than plaintext passwords
- Adversary: Interrogative
- Break: Universal forgery
- Today: The token got longer...

“... 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 tried a better encryption algorithm, but hit a bug that we couldn't fix, so we implemented one that worked even though the architect in charge was fully aware of its short-comings. You must understand that I'm giving you my read on the situation since **I've joined WSJ.com just 5 weeks ago.**”

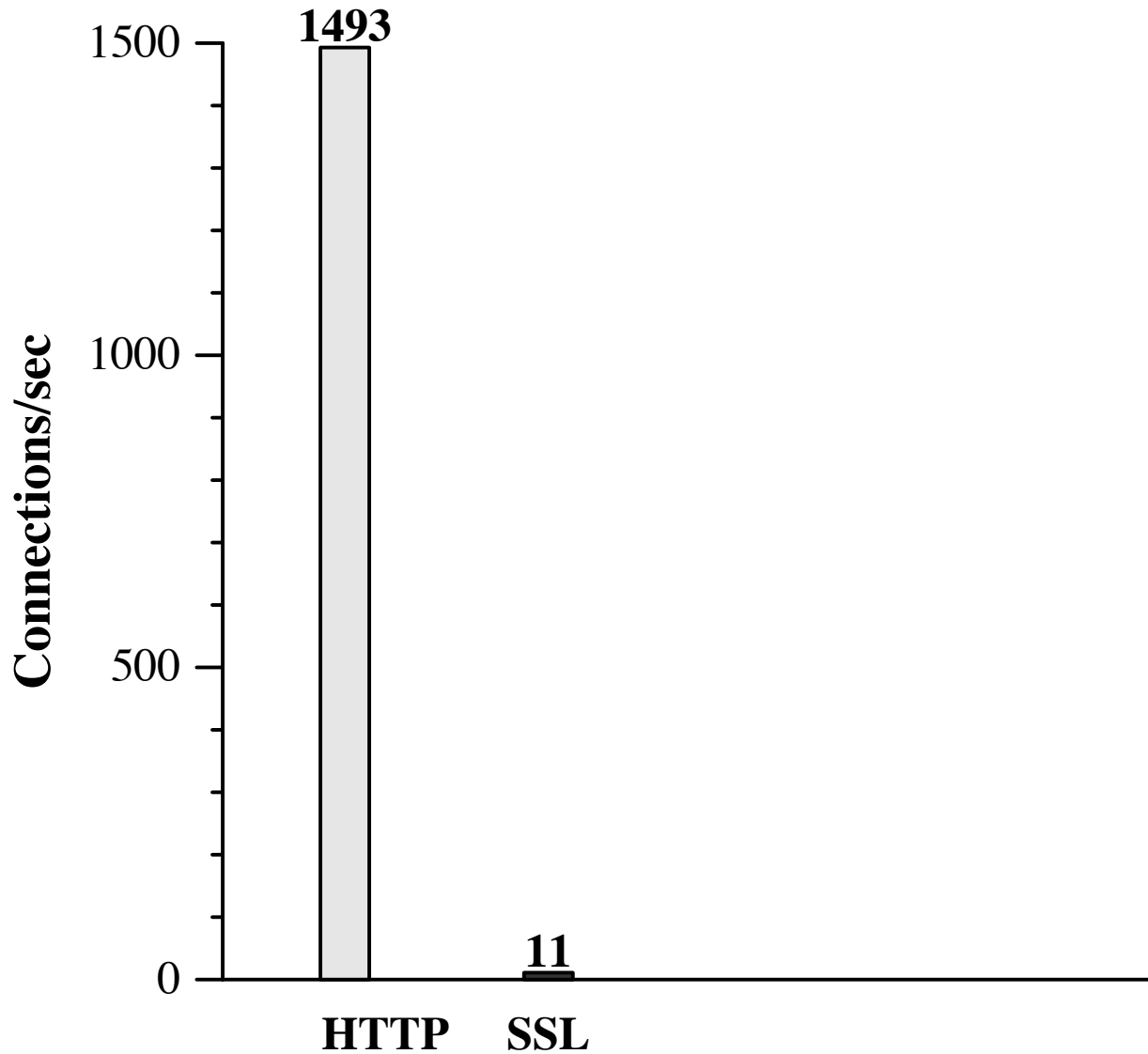
— Javeh Saleh, Vice President, Technology

Interactive Business Technology Services, WSJ.com

Why cookies?

- SSL is computationally expensive
- No one outside enterprises uses SSL client certificates
- Browsers offer an inflexible GUI for HTTP authentication
- Popular browsers implement cookies

HTTPS vs. HTTP handshake cost



How did we break these schemes?

- Gathered public information
 - Observe usernames and Web server HTTP responses
 - Obtain sample authenticators
 - Create guest accounts
- Observe authenticators while varying parameters
- No eavesdropping

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
- Encryption alone does not prevent forgery
- HighSchoolAlumni.com

Sign what you mean!

- badauth = sign (username + expiration, key)
 - (Alice, 21-Apr-2003)
 - sign (Alice21-Apr-2003, key)
 - (Alice2, 1-Apr-2003)
 - sign (Alice21-Apr-2003, key)
- Same authenticator!
“Alice” + “21-Apr-2003” ==
“Alice2” + “1-Apr-2003”
- Use unambiguous representation or delimiters

A scheme that mostly works

$$\text{auth} = \text{capa} + \text{expire} + \text{MAC}_k(\text{capa} + \text{expire})$$

**where MAC could be HMAC-SHA1,
capa could be an encrypted capability,
expire represents an encrypted expiration, and
'+' denotes concatenation with a delimiter**

Secure against **interrogative adversary**

A scheme that mostly works

auth = capa + expire + $\text{MAC}_k(\text{capa} + \text{expire})$

where MAC could be HMAC-SHA1,
capa could be an encrypted capability,
expire represents an encrypted expiration, and
'+' denotes concatenation with a delimiter

Secure against **interrogative** adversary

Still missing: A policy language for the
capability

The interrogative adversary defeats...

- SSL client authentication? No.
- HTTP Basic or Digest authentication? No.
- Homebrew cookie authentication schemes?
Often...

Vulnerability disclosure

- Vulnerability reporting is 1% technical analysis and 99% proper handling of disclosure.
- Report the bug to the vendor first. Then ask how long they need.
- There are release cycles and QA testing procedures. Be patient.
- Most companies are reasonable.
- If you are nice, you might get a free T-shirt. :-)

Summary

- Many schemes broken easily by the interrogative adversary
- Hints could prevent vulnerabilities
- There is a simple scheme that works
- Cookies are limited; live with it or move on