Client-side Defenses for Context-Aware Phishing and Transaction Generator Spyware

Collin Jackson

Dan Boneh John Mitchell

Stanford University

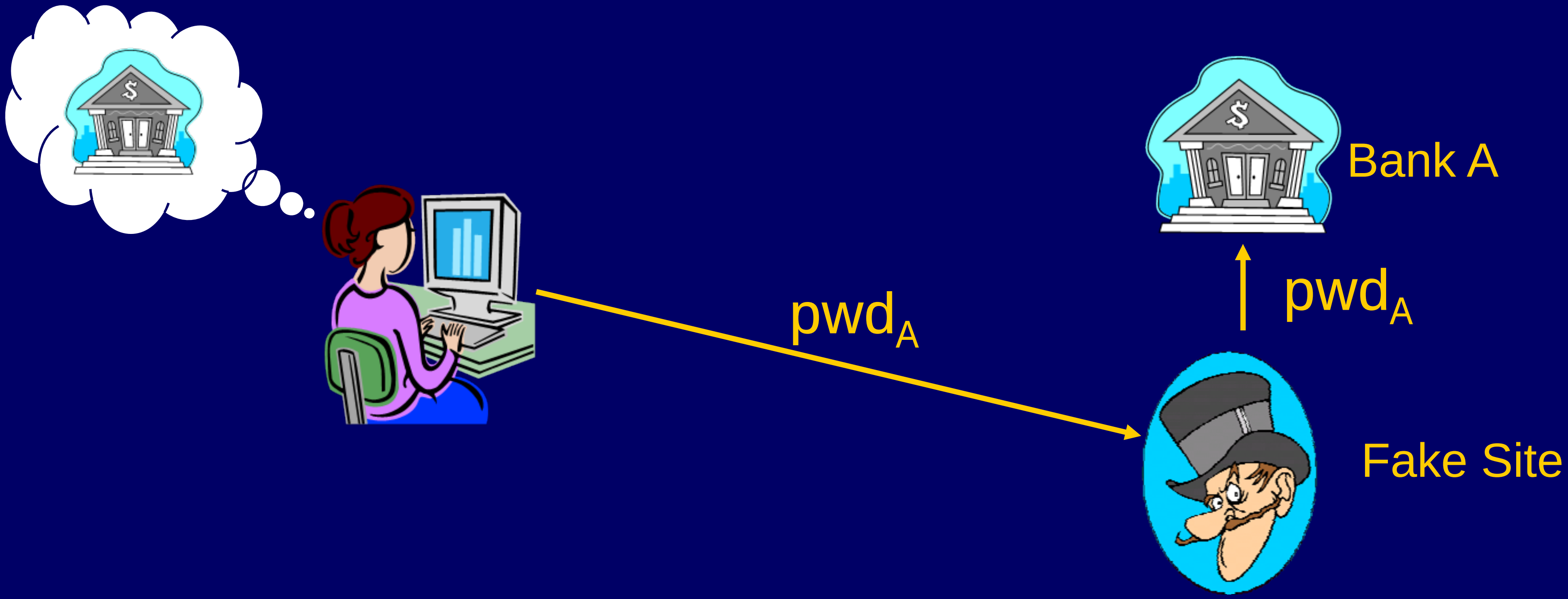
Web Threats

- Phishing
  - Spoof website convinces user to log in
- Common password problem
  - Same password for different sites
- Keylogger spyware
  - Malicious software observes login
- Transaction generator spyware
  - Hijacks login session, sends requests

Web Threats

- Phishing
  - SafeHistory
  - SpoofGuard
  - SafeCache
- Common password problem
  - PwdHash
- Keylogger spyware
  - SpyBlock (no server changes)
- Transaction generator spyware
  - SpyBlock (with server changes)

Phishing Problem



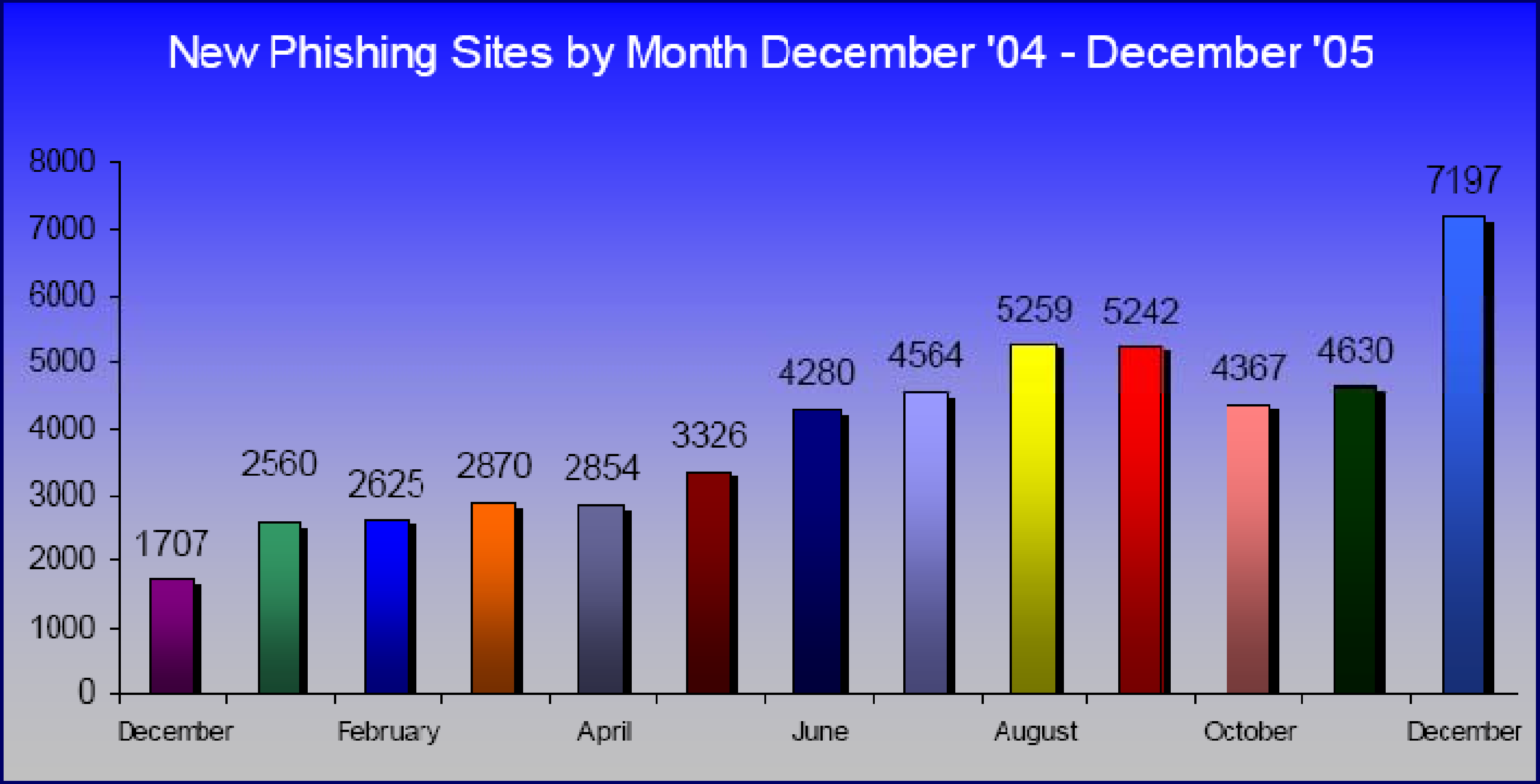
pwd$_A$

Bank A

pwd$_A$

Fake Site

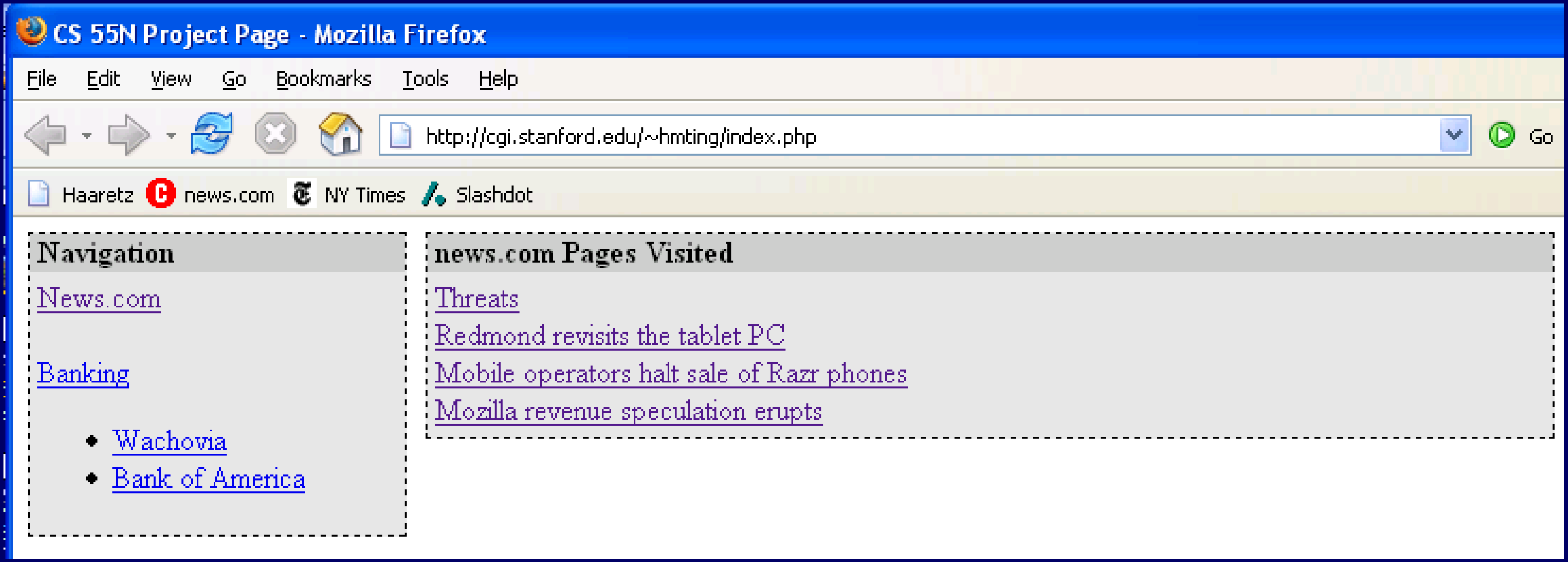
- User has existing relationship with target site
- User cannot reliably identify fake site
- Captured password can be used at target site



Anti-Phishing Working Group: antiphishing.org

Context-aware phishing

- Bank of America customers see:
  - "Please log in to your Bank of America account"
- Wells Fargo customers see:
  - "Please log in to your Wells Fargo account"
- Works in all major browsers, Outlook 2002


CS 55N Project Page - Mozilla Firefox
File
Edit
View
Go
Bookmarks
Tools
Help
http://cgi.stanford.edu/~hmting/index.php
Go
Haaretz
news.com
NY Times
Slashdot

Navigation

News.com

Banking

- Wachovia
- Bank of America

news.com Pages Visited

Threats
Redmond revisits the tablet PC
Mobile operators halt sale of Razr phones
Mozilla revenue speculation erupts

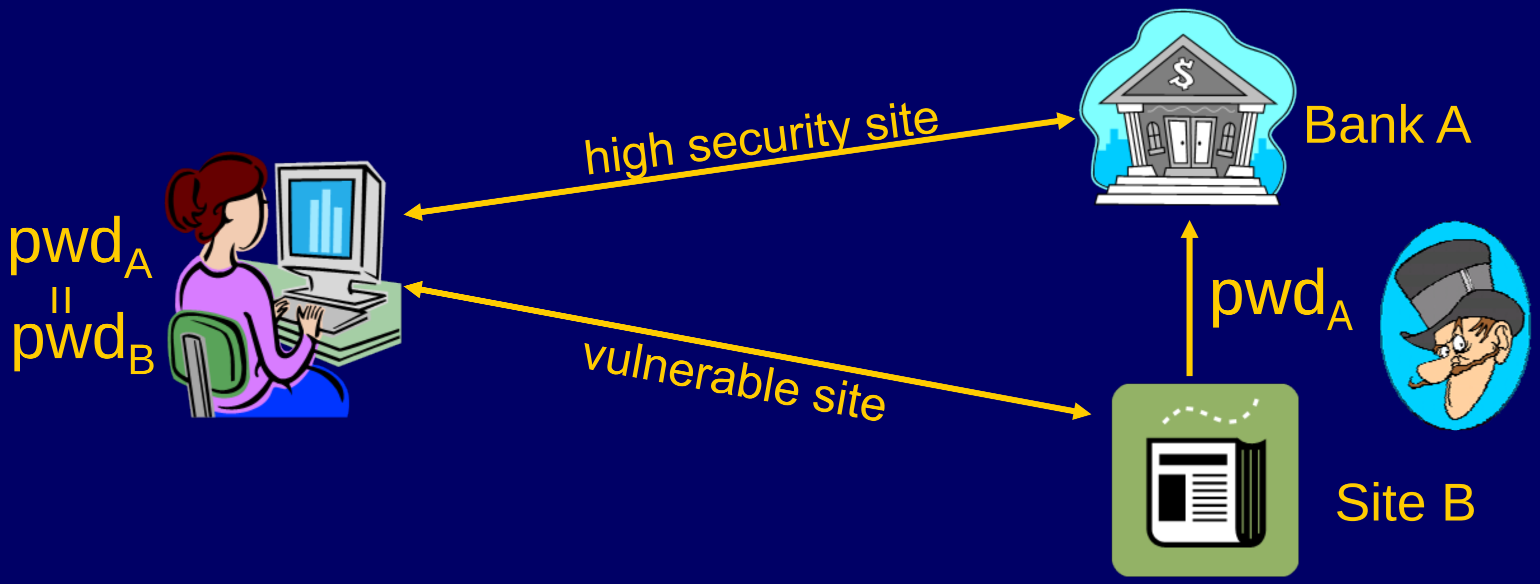
Protecting Browser State

C. Jackson, A. Bortz, D. Boneh, J. C. Mitchell (WWW ’06)

- Snooping violates same-origin principle:

  Only the site that stores some information in the browser may later read or modify that information.
- Stylesheets applied to hyperlinks
  - SafeHistory narrows policy to safe cases
- Javascript cache timing techniques
  - SafeCache partitions cache for safety

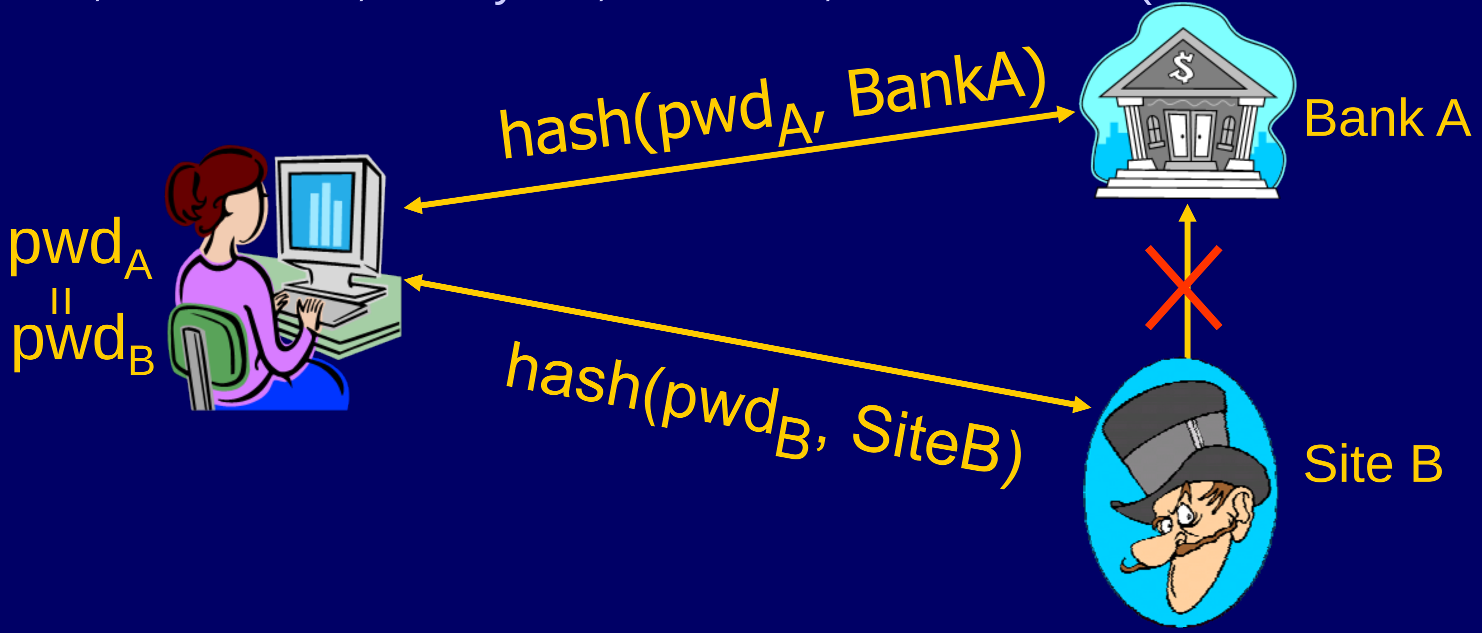
Common Password Problem



- Phishing attack or break-in at site B reveals pwd at A
  - Server-side solutions will not keep pwd safe
  - Solution: Strengthen with client-side support

PwdHash

B. Ross, C. Jackson, N. Miyake, D. Boneh, J. C. Mitchell (USENIX Sec ’05)



$\text{hash}(\text{pwd}_A, \text{BankA})$

Bank A

pwd_A
$=$
pwd_B

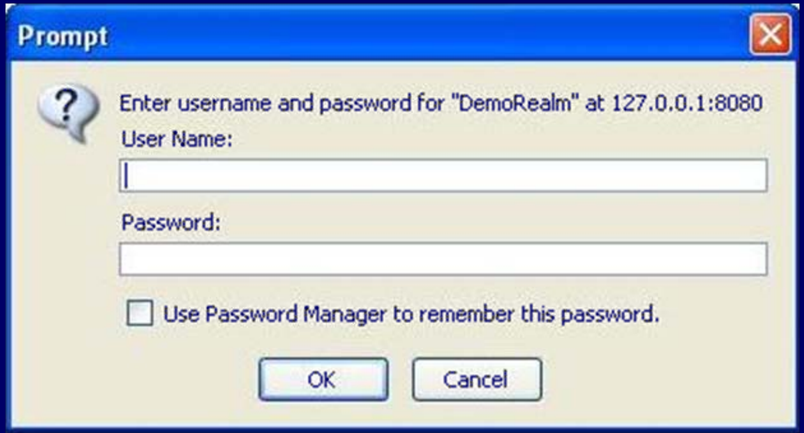
$\text{hash}(\text{pwd}_B, \text{SiteB})$

Site B

- Generate a unique password per site
  - $\text{HMAC}_{\text{fido:123}}$(banka.com) ⇒ Q7a+0ekEXb
  - $\text{HMAC}_{\text{fido:123}}$(siteb.com) ⇒ OzX2+ICiqc
- Hashed password is not usable at target site

User Interface Spoofing

- Attacker can display password fields or dialogs:



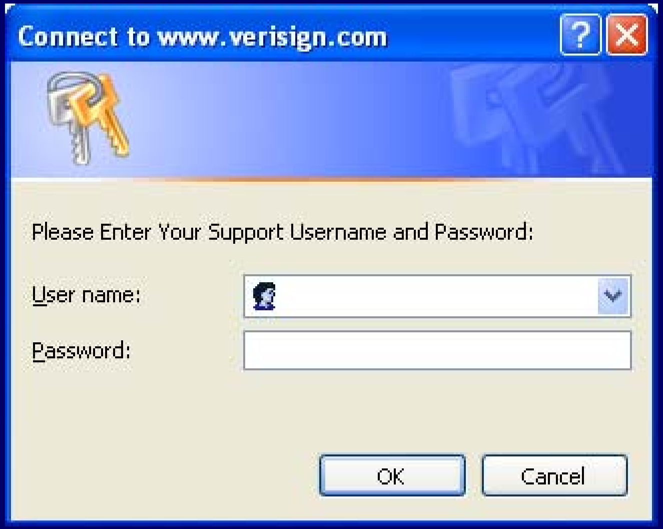
Prompt

Enter username and password for "DemoRealm" at 127.0.0.1:8080

User Name:

Password:

☐ Use Password Manager to remember this password.



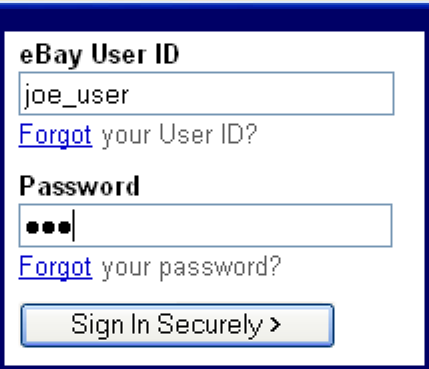
Connect to www.verisign.com

Please Enter Your Support Username and Password:

User name:

Password:

- Password is sent to attacker in the clear



eBay User ID

joe_user

Forgot your User ID?

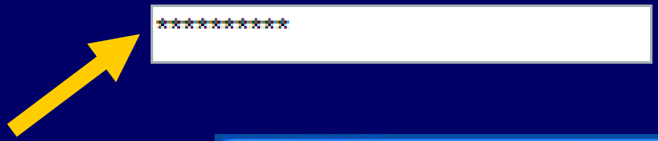
Password

●●●

Forgot your password?

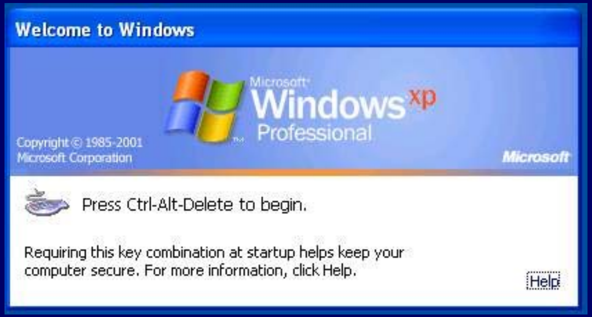
Trusted Password Interfaces

- Password prefix
  - PwdHash
- Secure attention sequence
- Isolated screen area
- Trusted image or phrase
  - Passmark
  - SpyBlock



****

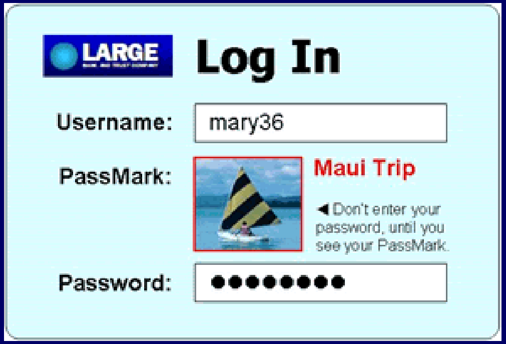
Starts with @@



Welcome to Windows

Press Ctrl-Alt-Delete to begin.

Requiring this key combination at startup helps keep your computer secure. For more information, click Help.



Log In

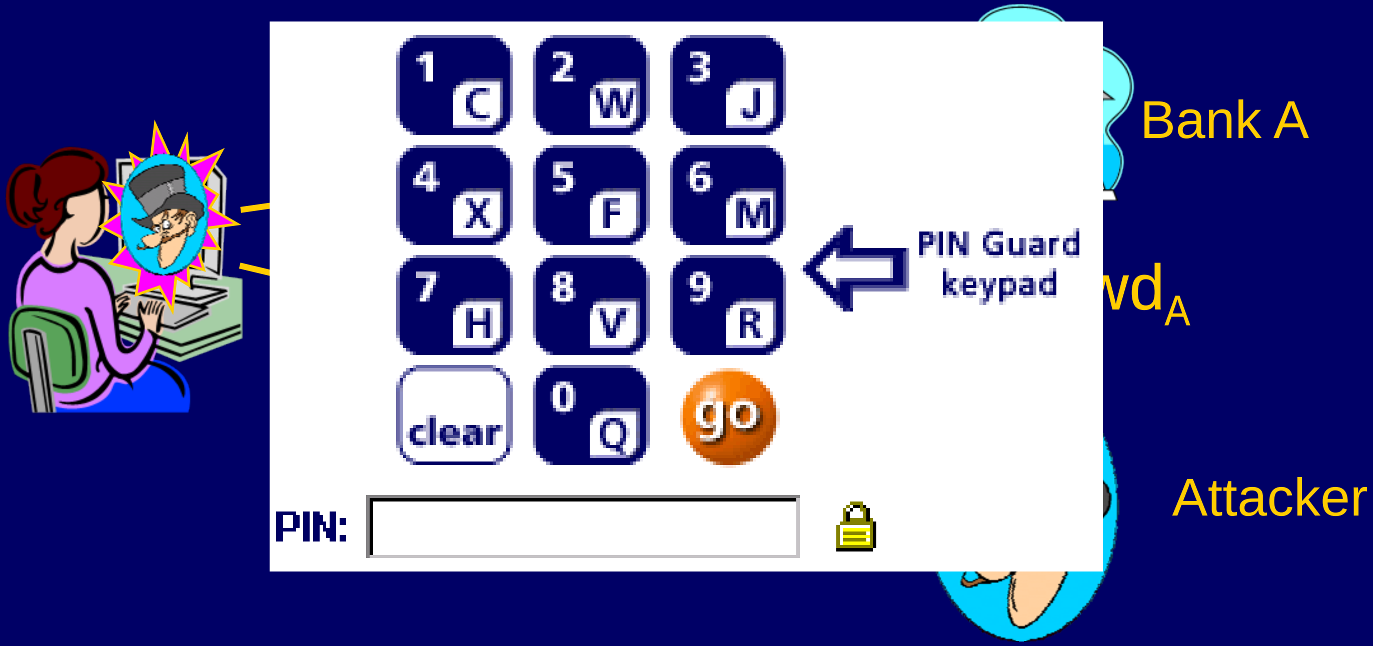
Username: mary36

PassMark: Maui Trip

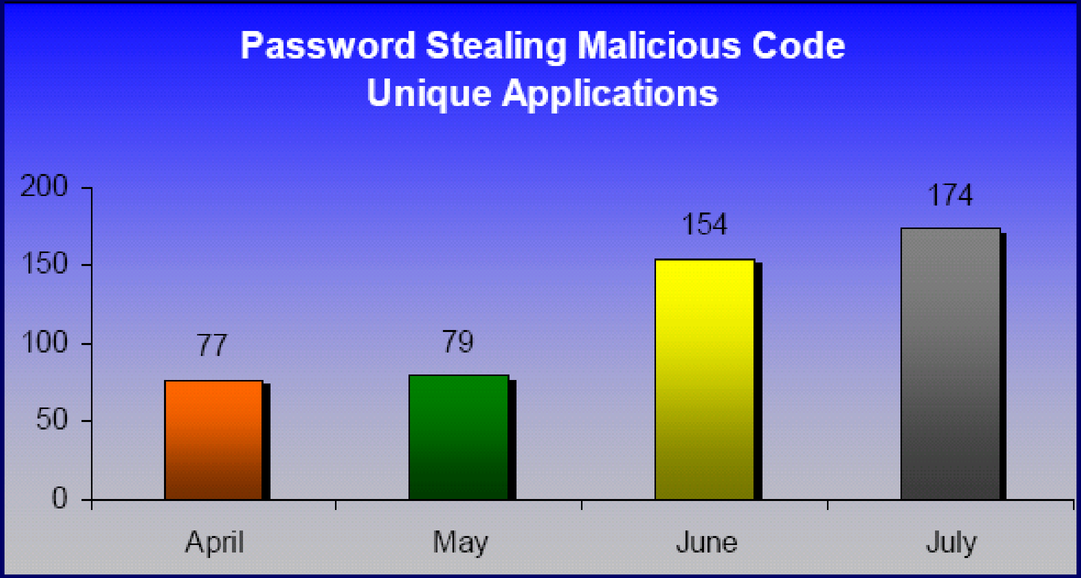
◀ Don't enter your password, until you see your PassMark.

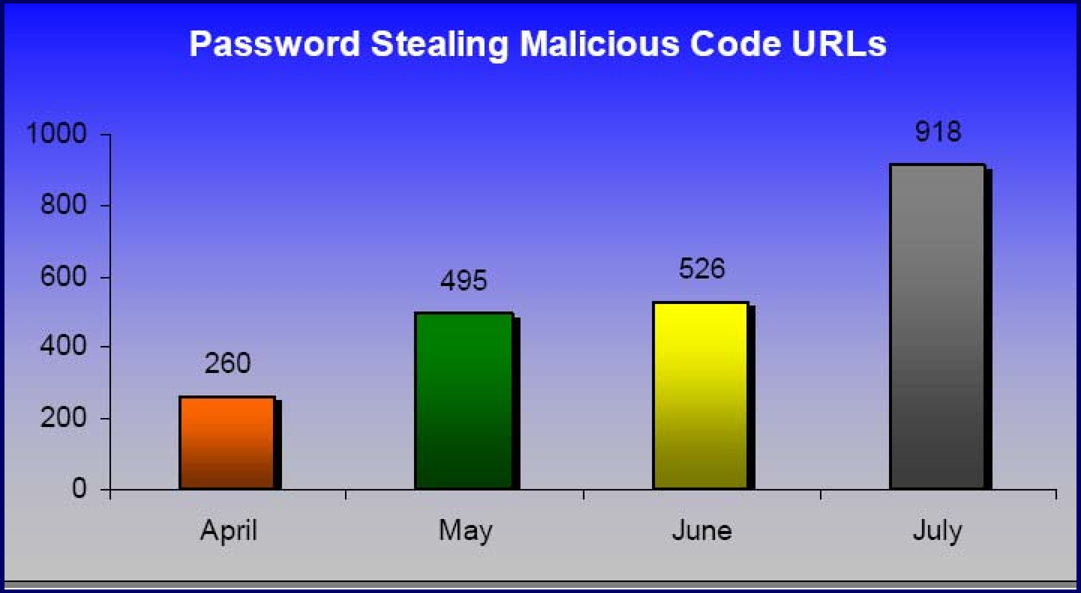
Password: ●●●●●●●●

Keylogger Spyware Problem



- Attacker observes login on local machine
- Password is sent to attacker for later use
- Screenshot can observe “screen keyboards”

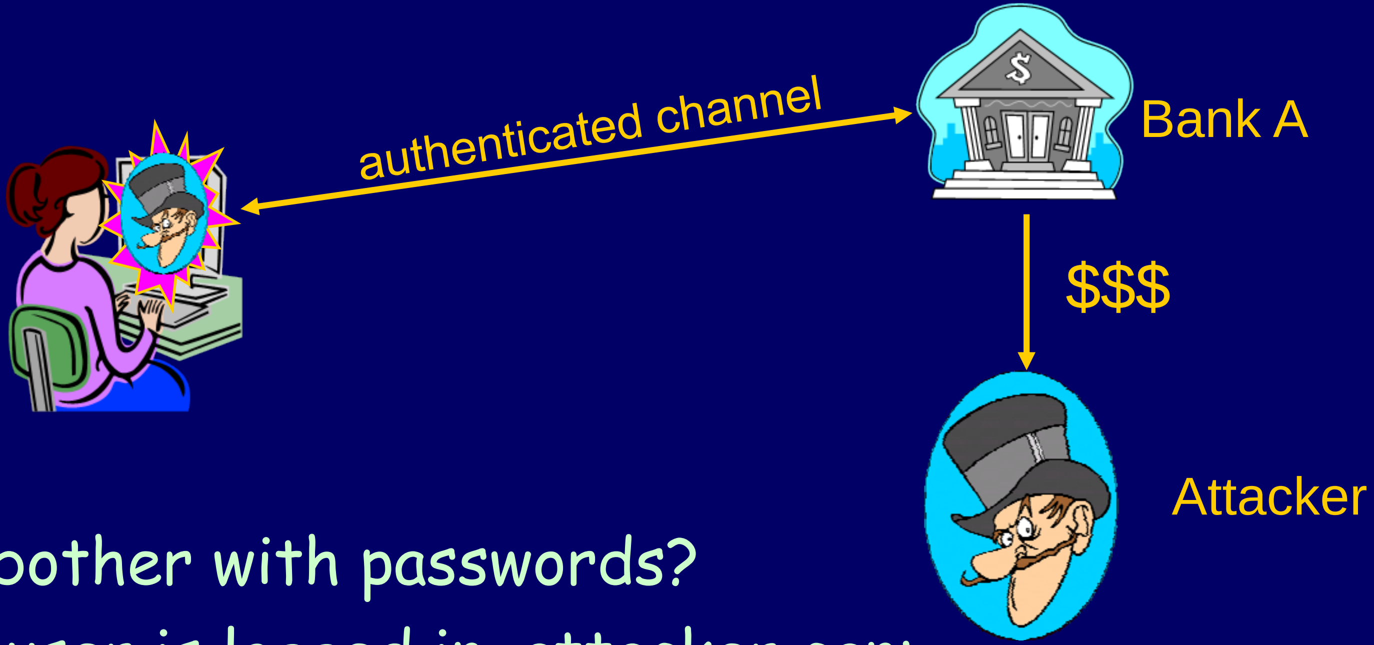

Password Stealing Malicious Code
Unique Applications
200
150
100
50
0
77
79
154
174
April
May
June
July


Password Stealing Malicious Code URLs
1000
800
600
400
200
0
260
495
526
918
April
May
June
July

APWG

July '05

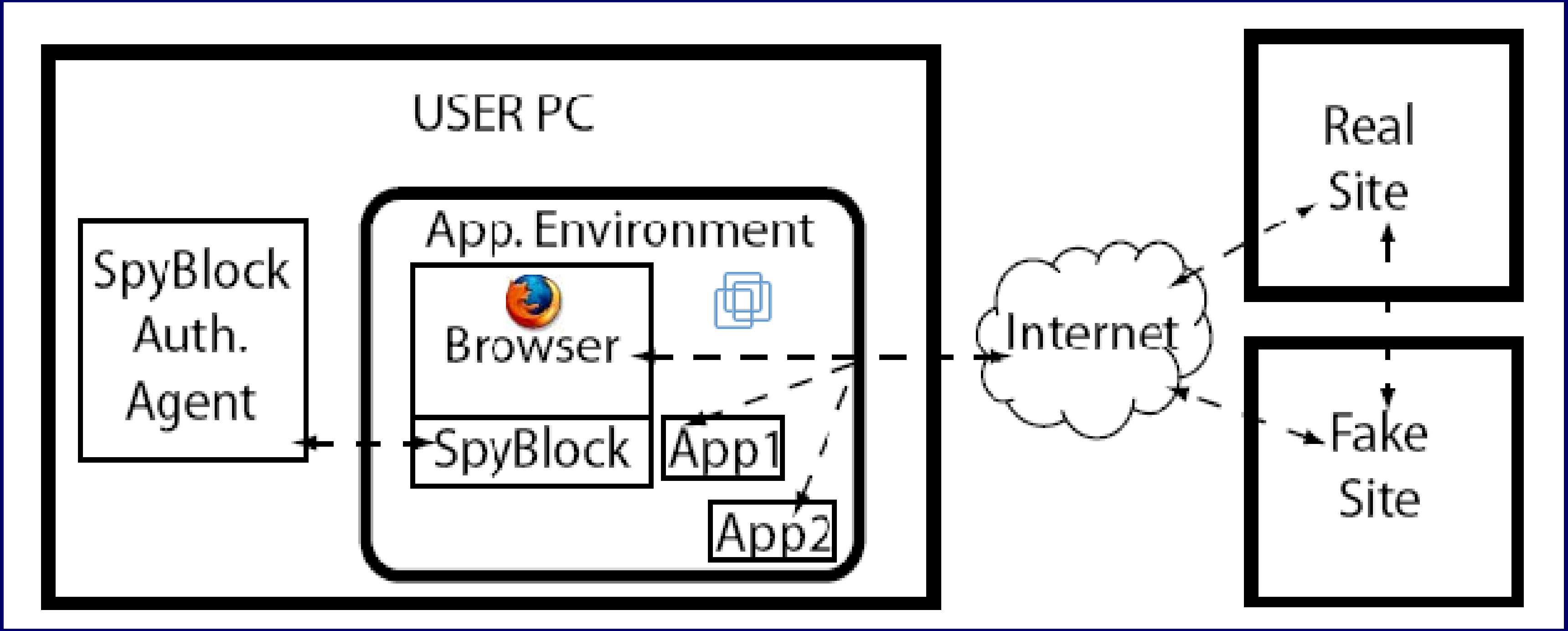
Transaction Generator Problem



- Why bother with passwords?
- Once user is logged in, attacker can:
  - Corrupt user requests
  - Issue unauthorized requests

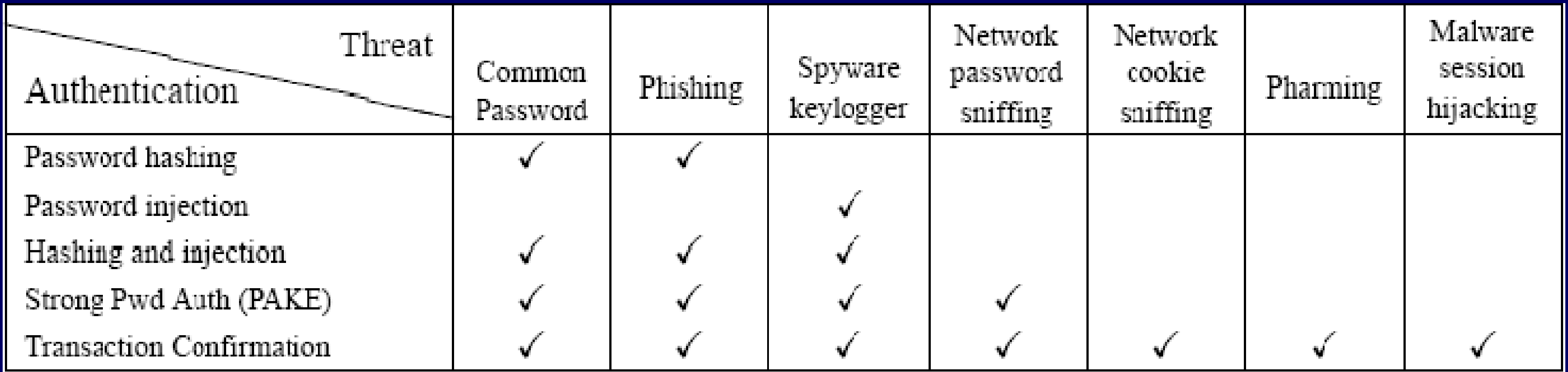
SpyBlock

C. Jackson, D. Boneh, J. C. Mitchell



- Isolated component for authentication
- Untrusted environment for user apps

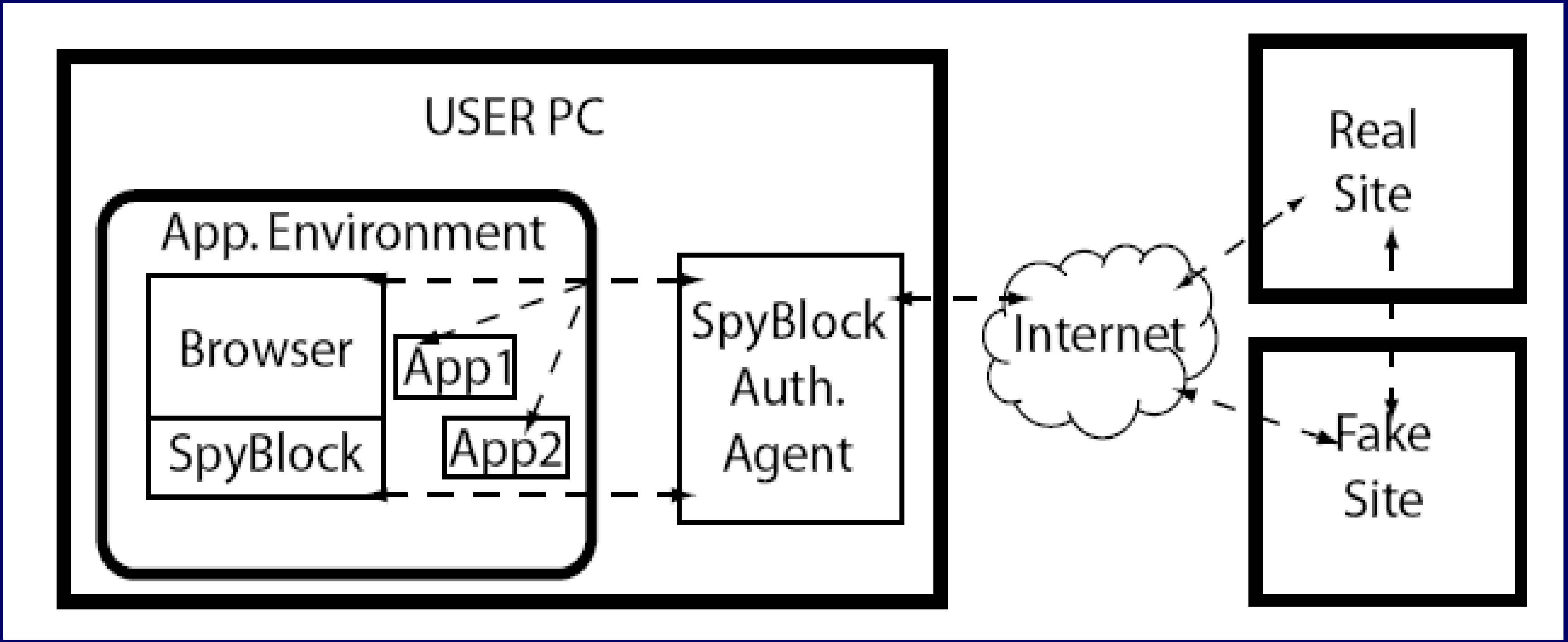
Authentication modes



Threat Authentication	Common Password	Phishing	Spyware keylogger	Network password sniffing	Network cookie sniffing	Pharming	Malware session hijacking
Password hashing	✓	✓					
Password injection			✓				
Hashing and injection	✓	✓	✓				
Strong Pwd Auth (PAKE)	✓	✓	✓	✓			
Transaction Confirmation	✓	✓	✓	✓	✓	✓	✓

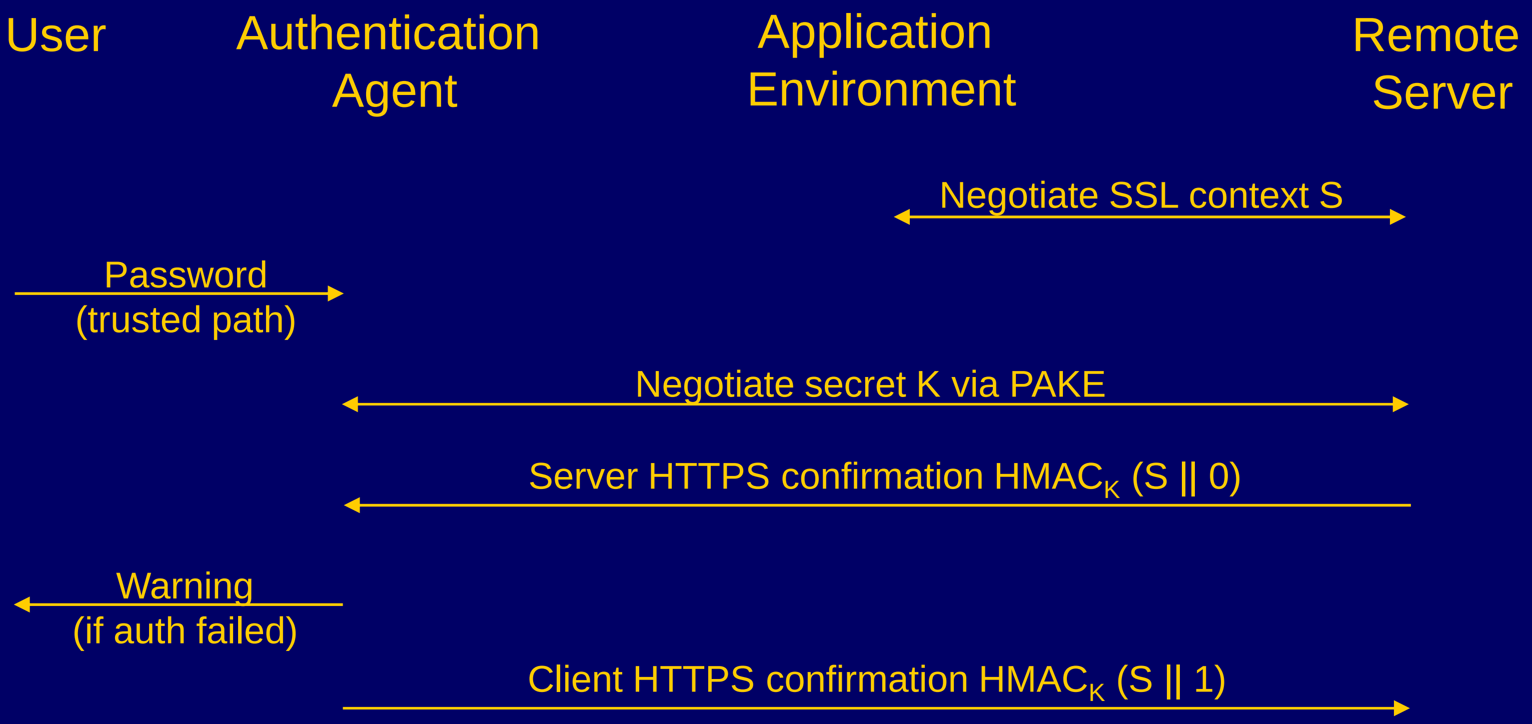
- Hashing, injection require no server assistance
- Server support for additional protection

Password injection



- Intercept outbound requests and insert password
- Check for password fields in HTML to deter reflection

Strong password authentication



User

Authentication Agent

Application Environment

Remote Server

Negotiate SSL context S

Password (trusted path)

Negotiate secret K via PAKE

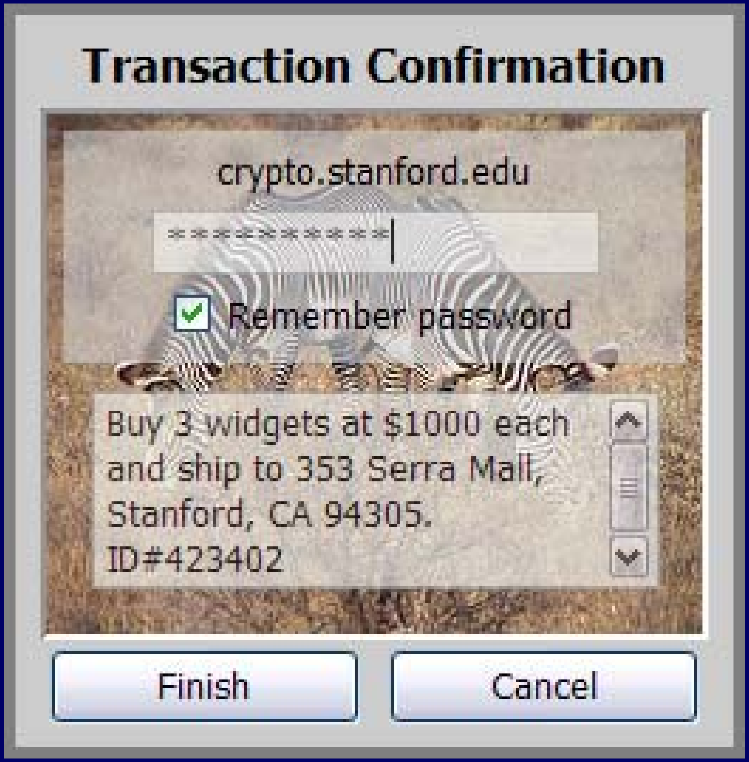
Server HTTPS confirmation HMAC_K (S || 0)

Warning (if auth failed)

Client HTTPS confirmation HMAC_K (S || 1)

- Application environment does not learn user password
- HTTPS is verified to prevent network man-in-the-middle

Transaction confirmation



Transaction Confirmation

crypto.stanford.edu

****

☑ Remember password

Buy 3 widgets at $1000 each and ship to 353 Serra Mall, Stanford, CA 94305. ID#423402

- Application environment cannot MAC fake transaction
- Unique transaction ID prevents replay attacks

Project websites

- Phishing

  www.safehistory.com

  www.safecache.com

- Common password problem

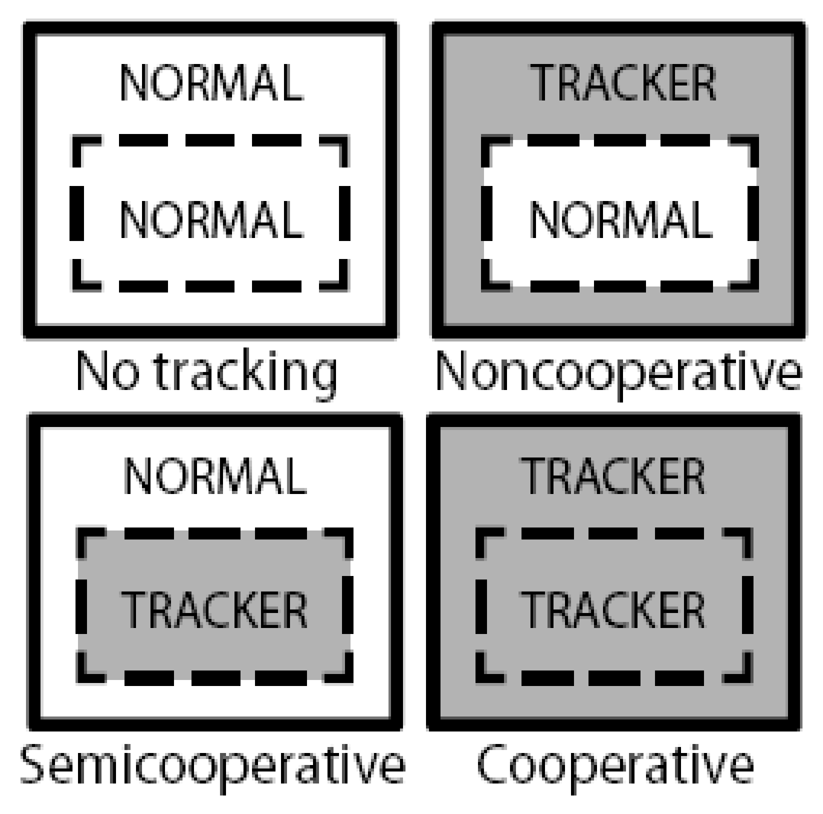
  www.pwdhash.com

- Keylogger spyware
- Transaction generator spyware

  www.getspyblock.com

Browser Access Control

- Noncooperative:
  - Same origin policy
- Semicooperative:
  - Third party blocking policy
- Cooperative:
  - ???



Why use Password Prefix?

- Protection mechanism “built in” to password
- Does not rely on user to make a decision
- Same prefix works for everyone
- Distinguishes secure passwords from
  - normal passwords
  - social security numbers
  - PINs
- Only use it when you want to

Why use Password Prefix?

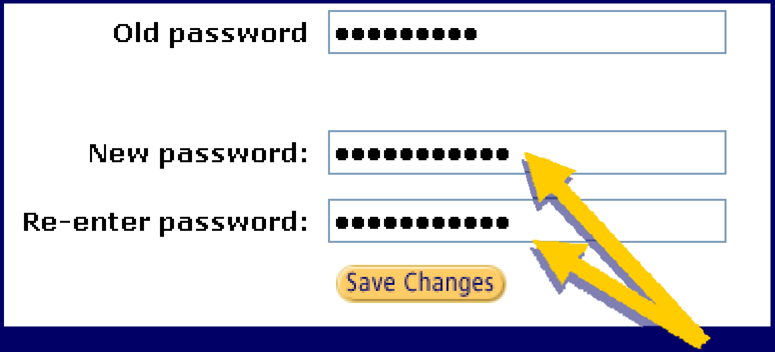
- Protection mechanism “built in” to password
- Does not rely on user to make a decision
- Same prefix works for everyone
- Distinguishes secure passwords from
  - normal passwords
  - social security numbers
  - PINs
- Only use it when you want to

Other Challenges

- Password Reset
- Internet Cafes
- Dictionary Attacks
- Spyware, DNS poisoning (no protection)
- Other issues (described in the paper)
  - Choosing salt for hash
  - Encoding hashed password
  - Additional attacks and defenses

Password Reset

- After install, PwdHash can't protect existing pwds
  - Only passwords starting with @@ are secure
  - User can choose where to use PwdHash
  - User must enter old password unhashed into password reset page
- Pwd Prefix makes it easy
  - Old passwords won't be accidentally hashed
  - New, secure passwords are automatically hashed



Old password ●●●●●●●●●

New password: ●●●●●●●●●●●

Re-enter password: ●●●●●●●●●●●

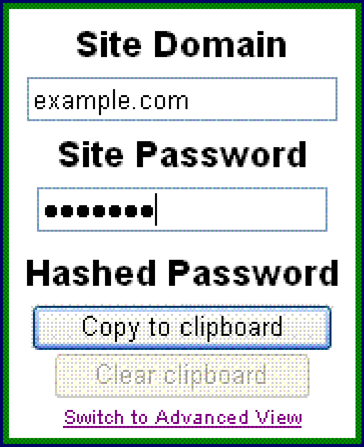
Starts with @@

Internet Cafes

- Users cannot install software at Internet Cafes.
- Would not be a problem if PwdHash were universally available
- Interim solution: A secure web site for remote hashing, e.g.

  https://www.pwdhash.com

- Hash is computed using JavaScript
  - Server never sees password
  - Resulting hash is copied into clipboard
  - Can also be used as a standalone password generator

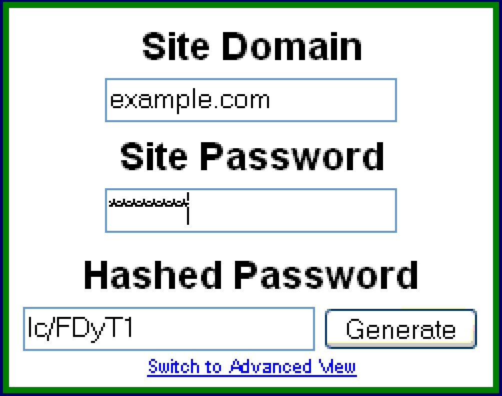


Site Domain: example.com

Site Password: ●●●●●●●

Hashed Password

Internet Explorer



Site Domain: example.com

Site Password: ********

Hashed Password: Ic/FDyT1

Firefox

Dictionary attacks

- After phishing attack or break-in to low security site, attacker can repeatedly guess password and check hash.
  - Succeeds on ≈15% of passwords (unlike 100% today)
  - Less effective on longer, stronger passwords
- Solution: better authentication protocol (SPEKE, SRP, etc.)
  - Requires server-side changes
- Defense: user specifies a global pwd to strengthen all pwd hashes
  - Creates a new pwd management problem for shared machines
- Defense: slow hash function (Halderman, Waters, Felten '05)
  - Increases time of dictionary attack