

# CS 294-28 : INTERNET/NETWORK SECURITY

---

## **Stronger Password Authentication Using Browser Extensions**

*--B. Ross, C. Jackson, N. Miyake, D. Boneh, and J.C. Mitchell, Stanford*

## **PwdIP-Hash: A Lightweight Solution to Phishing and Pharming Attacks**

*--Barber Aslam, Lei Wu, and Cliff C. Zou, Univ of Central Florida*

Presented by: Gabriel Núñez (some slides borrowed from authors)

# USER BEHAVIOR

- Security and Privacy decisions

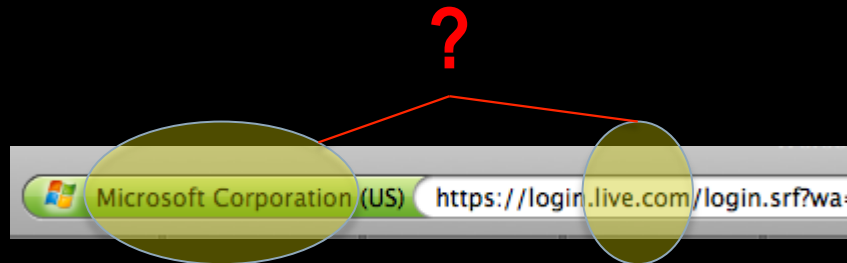
- Protocol (https://)

- Domain name

- SSL Certificate

- Visual elements

- Lock symbol/icon
- Extended validation (EV) certificates



# PROBLEMS

- Should be checked before entering data
    - Every visit to each site
  - Indicators can be spoofed or removed
  - Users in control
    - Education and expertise vary
  - Users ignore and/or don't understand warnings
-

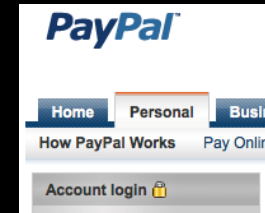
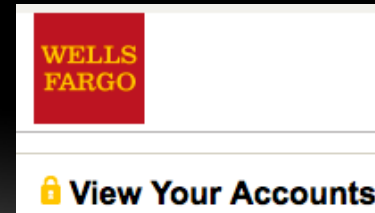
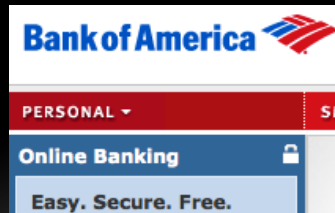
# STUDY (BY DOUGLAS STEBILA)

- Determine why users fail to follow security indicators
- “Users do poorly at understanding security indicators because they are effectively trained to do so by website designers.”

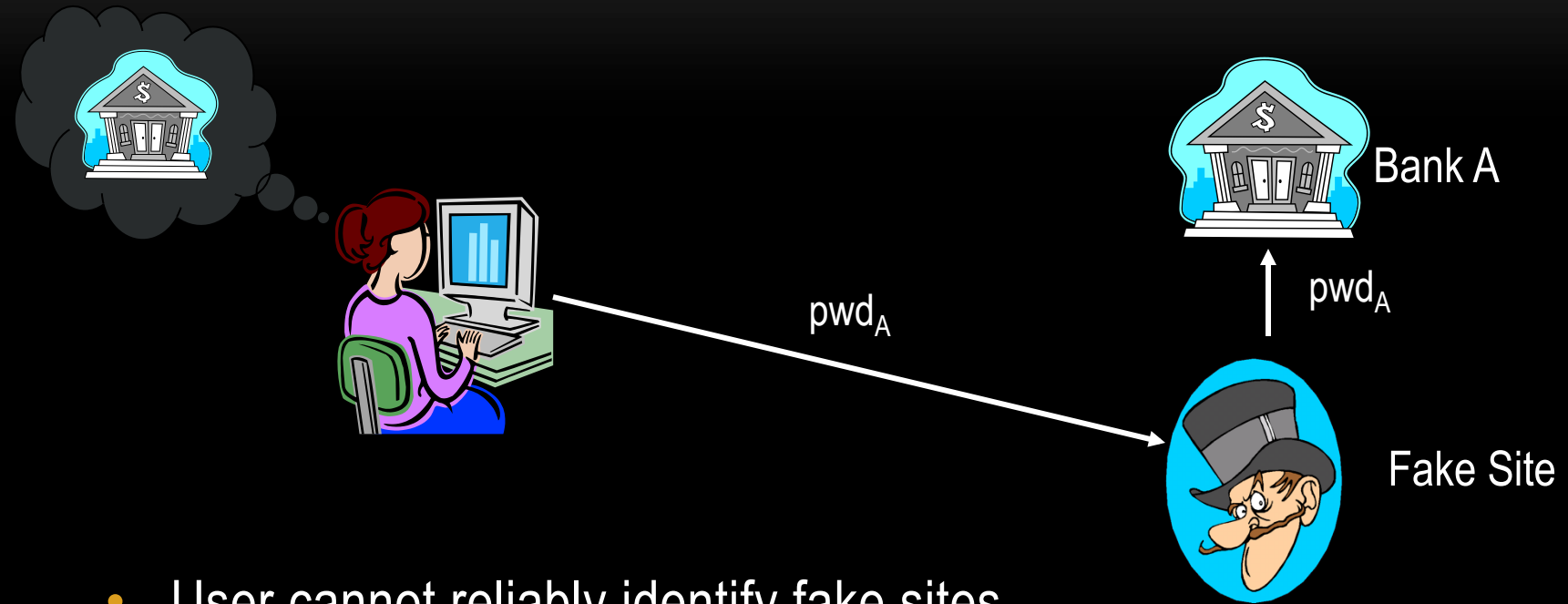
Misused security indicator	<i>n</i>
HTTP login page with HTTPS form submission	19
Lock icon on page or as favicon	29
Hidden location bar	2
Mismatched domain name	16
Very complicated URL	53

Login page	Form submission	<i>n</i>
	HTTP	56
HTTP	HTTPS	13
	HTTPS w/EV cert.	6
HTTPS	HTTPS	40
HTTPS w/EV cert.	HTTPS w/EV cert.	10

Match?	Example typed domain → login domain	<i>n</i>
Exact match	google.com → www.google.com	71
Close match	yahoo.com → login.yahoo.com	38
No match	hotmail.com → login.live.com	16

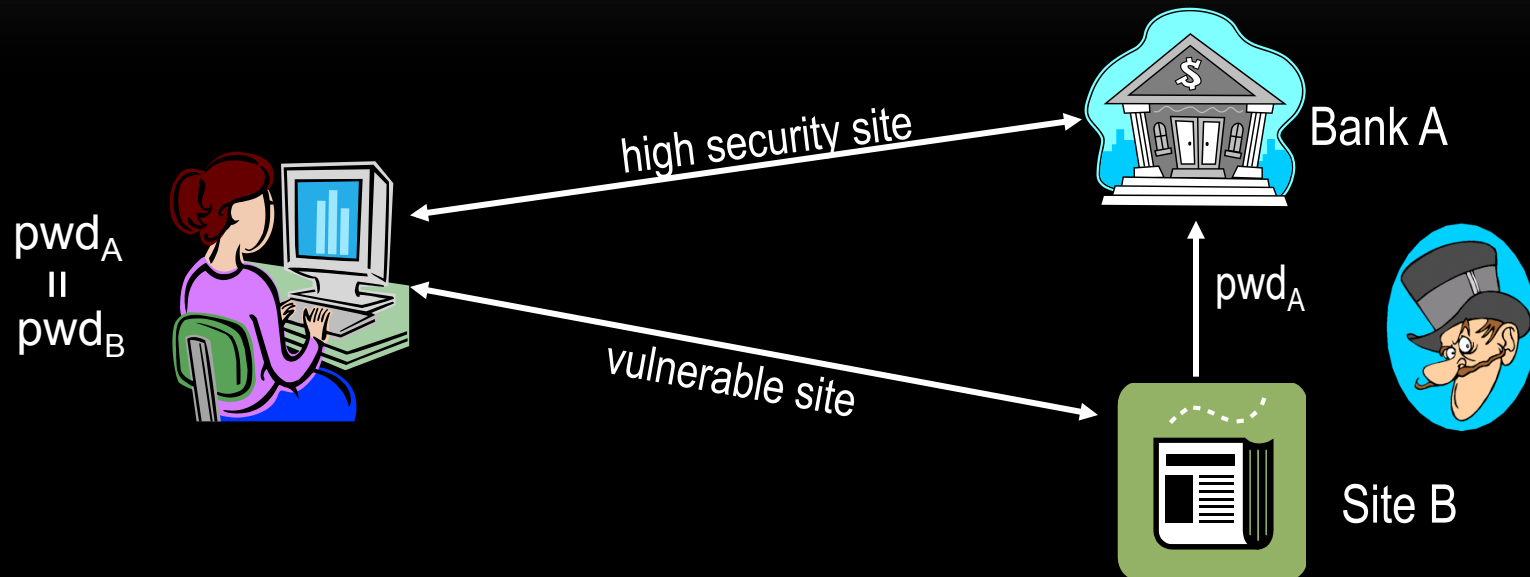


# PASSWORD PHISHING PROBLEM



- User cannot reliably identify fake sites
- Captured password can be used at target site

# COMMON PASSWORD PROBLEM

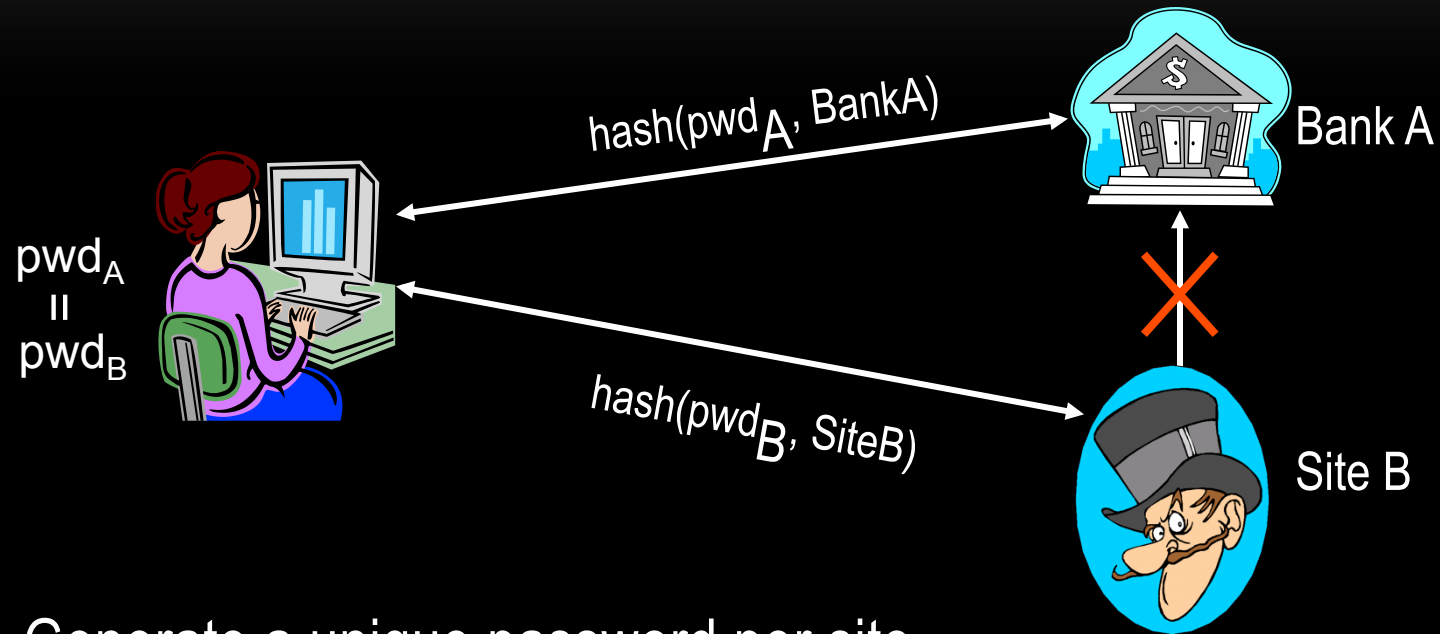


- Phishing attack or break-in at site B reveals  $\text{pwd}$  at A
  - Server-side solutions will not keep  $\text{pwd}$  safe
  - Solution: Strengthen with client-side support

# SOLUTION: PWDHASH

- Lightweight browser extension
- Impedes password theft
- Invisible to server
  - Pwd hashing
- Invisible to user
  - Pwd prefix

# PASSWORD HASHING



- Generate a unique password per site
  - $\text{HMAC}_{\text{fido:123}}(\text{banka.com}) \Rightarrow \text{Q7a+0ekEXb}$
  - $\text{HMAC}_{\text{fido:123}}(\text{siteb.com}) \Rightarrow \text{OzX2+ICiqc}$

# PWDHASH BENEFITS

- Triggering “password mode”
  - Password-key (F2-password)
  - Password-prefix (@@password)
- Compatible with auto-complete
  - Saves @@password or hashed password
- Roaming? No problem!
  - <https://www.pwdhash.com/>
  - JavaScript generates hash
    - Password never sent to server

# PASSWORD-MODE RESISTS JAVASCRIPT ATTACKS

- Keyloggers
    - User types @@password
    - Browser sees @@abcdefgh
  - Fake password fields
    - Warn if trigger password-mode while not in password field
  - Domain rewriting
  - Password reflection
-

# PROBLEMS WITH PWDHASH

- Partial implementation attack
  - 1<sup>st</sup> login attempt w/ @@password fails
  - 2<sup>nd</sup> attempt user tries normal password
- $\text{hash}(\text{pwd}_A, \text{BankA}) = \text{pwd}_{\text{BankA}}$ 
  - Vulnerable to eavesdropping
  - D. Stebila: 45% sites provide no encryption!
    - 15% have HTTP login pages w/ secure submission
- “We believe that providing customized passwords...”

# PWDIP-HASH (A CHANGES-REQUIRED APPROACH)

- Password never sent!
- Extended hash function
  - Password
  - Domain name (or URL\*)
  - Nonce from server
  - Remote-server's IP address
- Server maintains no state
- Puzzle for user

1. C     Setup TCP Connection
2. S     Generate  $N_S$ , compute  $E_K^- \{ N_S \}$
3. S→C:  $E_K^- \{ N_S \}, Cert_S$
4. C     Compute  $K = H_K^+ ( Dom | N_S | IP_S ); P_H = H_K ( P ); E_K^+ \{ P_H \}$
5. C→S:  $E_K^+ \{ P_H \}, E_K^- \{ N_S \}$
6. S     Compute  $K = H_K^+ ( Dom | N_S | IP_S ); P_{HS} = H_K ( P );$   
Verify ( $P_{HS} = P_H$ )

TABLE I. NOTATIONS AND DESCRIPTIONS

Notations	Descriptions
$C$	Client/User
$S$	Server
$IP_S$	Server IP-address
$N_S$	Nonce generated by Server
$P_H$	Hash value of user-typed password $P$
$Cert_S$	Certificate of server $S$ , defined by its key pair $(K^+, K^-)$ .
$(K^+, K^-)$	Public and private key pair of server
$E_K\{M\}$	An encryption function on message $M$ using the key $K$
$H_K(M)$	A secure hash function using key $K$ on message $M$
$Dom$	Domain-name

\* = optional

THANK YOU

---