Secure Passwords Through Enhanced Hashing

Benjamin Strahs, Chuan Yue, and Haining Wang

The College of William and Mary

W&M Computer Science

WILLIAM AND MARY

Passwords

- The most common online authentication method
- Something you know instead of something you have (hardware token) or something you are (biometrics)
- Simple, inexpensive, and convenient
- Will remain dominant in the foreseeable future

Problems

- Weak passwords are easy to crack
  - Short, common, easy to guess (e.g., “secret”, “susan123”)
  - Vulnerable to brute-force and dictionary attacks
  - Users often choose weak passwords (easy to remember)
- Passwords are vulnerable to theft
  - Phishing, key logging, shoulder surfing, etc.

Even worse: more accounts, password sharing (6.5 over 25)

Techniques to Securing Passwords

- Password managers
  - Lack mobility
- Single sign-on systems
  - Single point of failure
- Graphic passwords
  - Not mature, security and usability concerns
- Password hashing
  - Usability concerns, but very promising

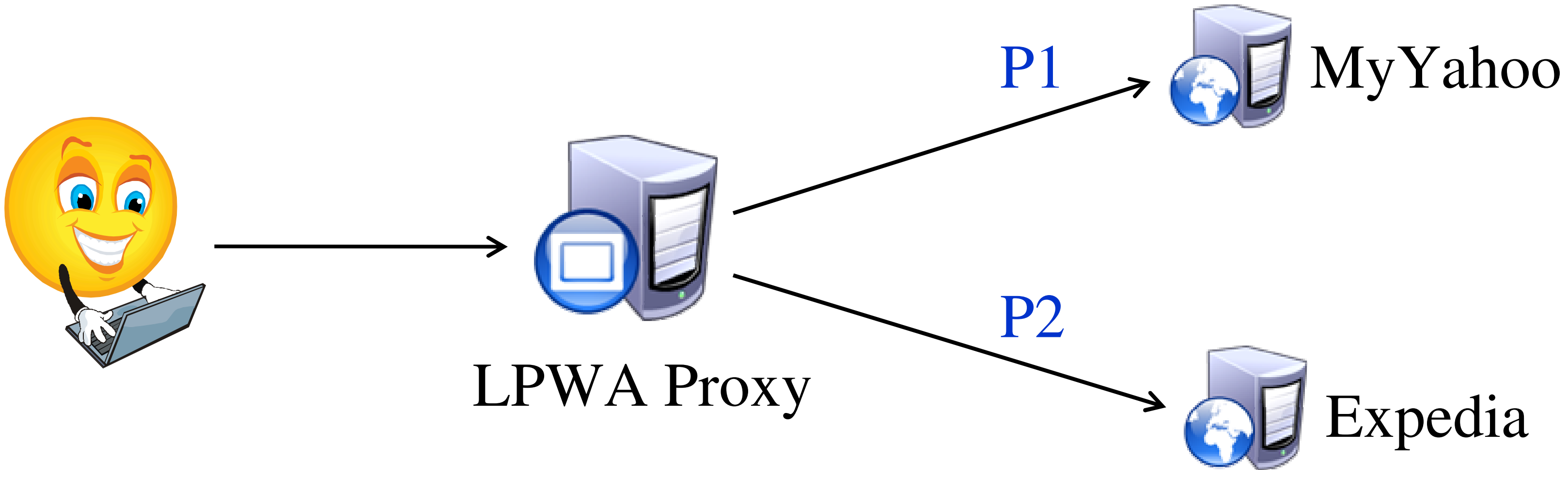
Outline

- Introduction
- Related work
- PasswordAgent
  - Design
  - Implementation
  - Evaluation
  - Limitations

Representative Hashing-based Systems

- LPWA (Lucent Personal Web Assistant)
  - Gabber et al., *Commun. ACM*, 1999
- PwdHash
  - Ross et al., *USENIX Security Symposium*, 2005
- Password Multiplier
  - Halderman,et al., *WWW*, 2005
- Passpet
  - Yee and Sitaker, *SOUPS*, 2006

Lucent Personal Web Assistant (LPWA)



P1

MyYahoo

LPWA Proxy

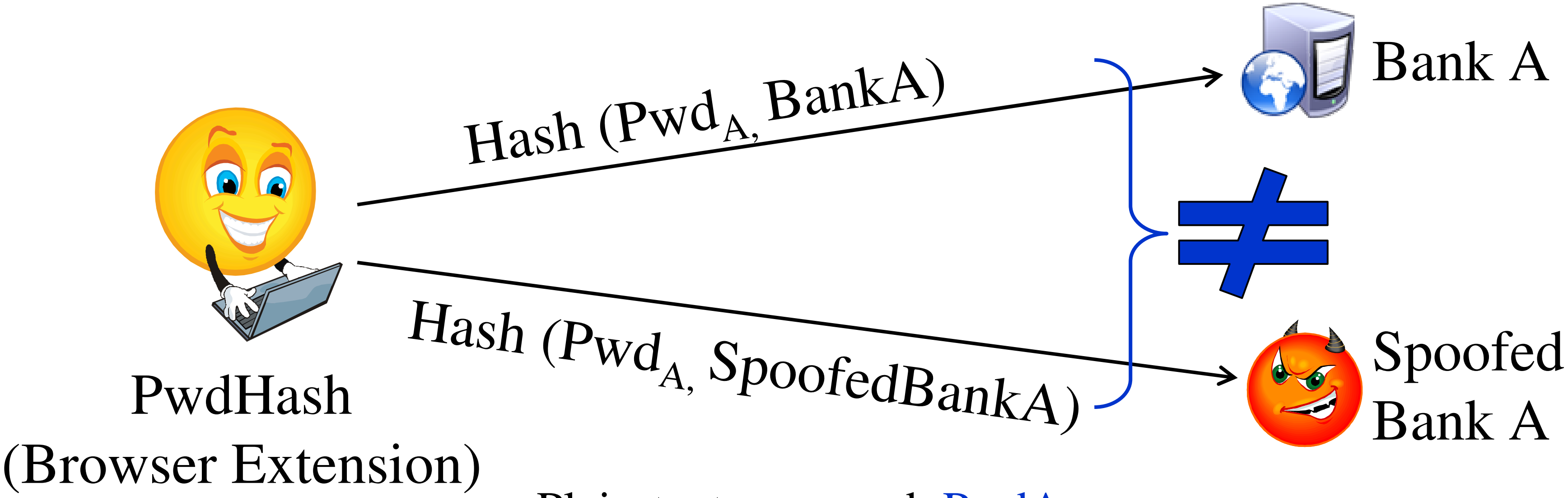
P2

Expedia

(Username, Password Website Address) hash → (Pseudonymous Username, Password, Email Address)

- Focuses on enabling anonymous Web access, anti-spam

PwdHash



Hash (Pwd$_{A,}$ BankA)

Bank A

≠

Hash (Pwd$_{A,}$ SpoofedBankA)

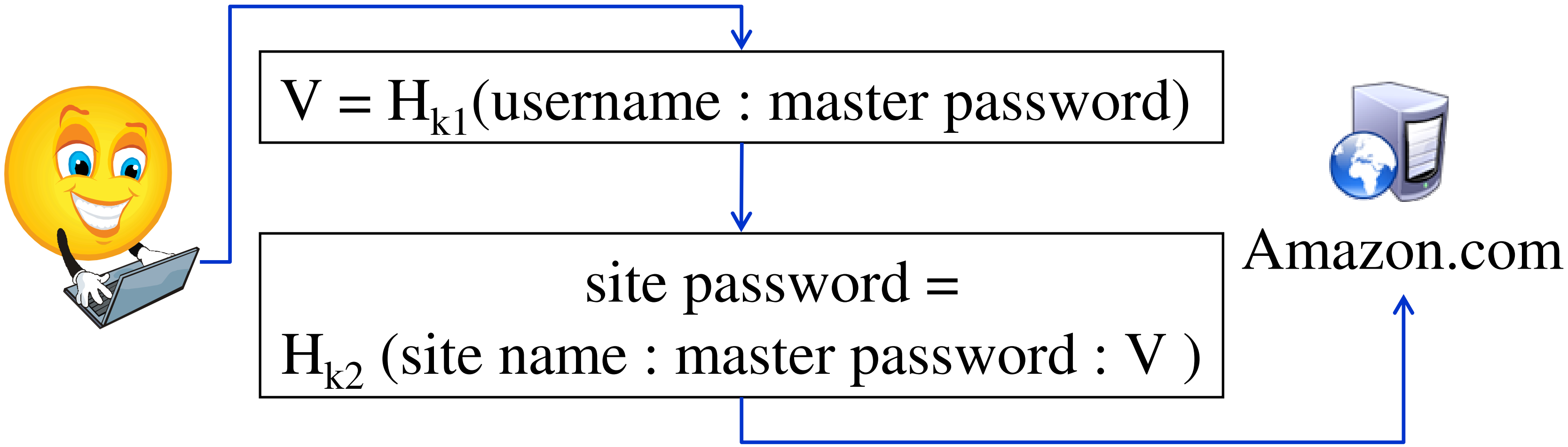
Spoofed Bank A

PwdHash
(Browser Extension)

Plain-text password: PwdA
Site-password: Hash (PwdA, BankA)

- Unique password per site (domain name is the salt)
- Focuses on protecting against phishing attacks

Password Multiplier



$V = H_{k1}$(username : master password)

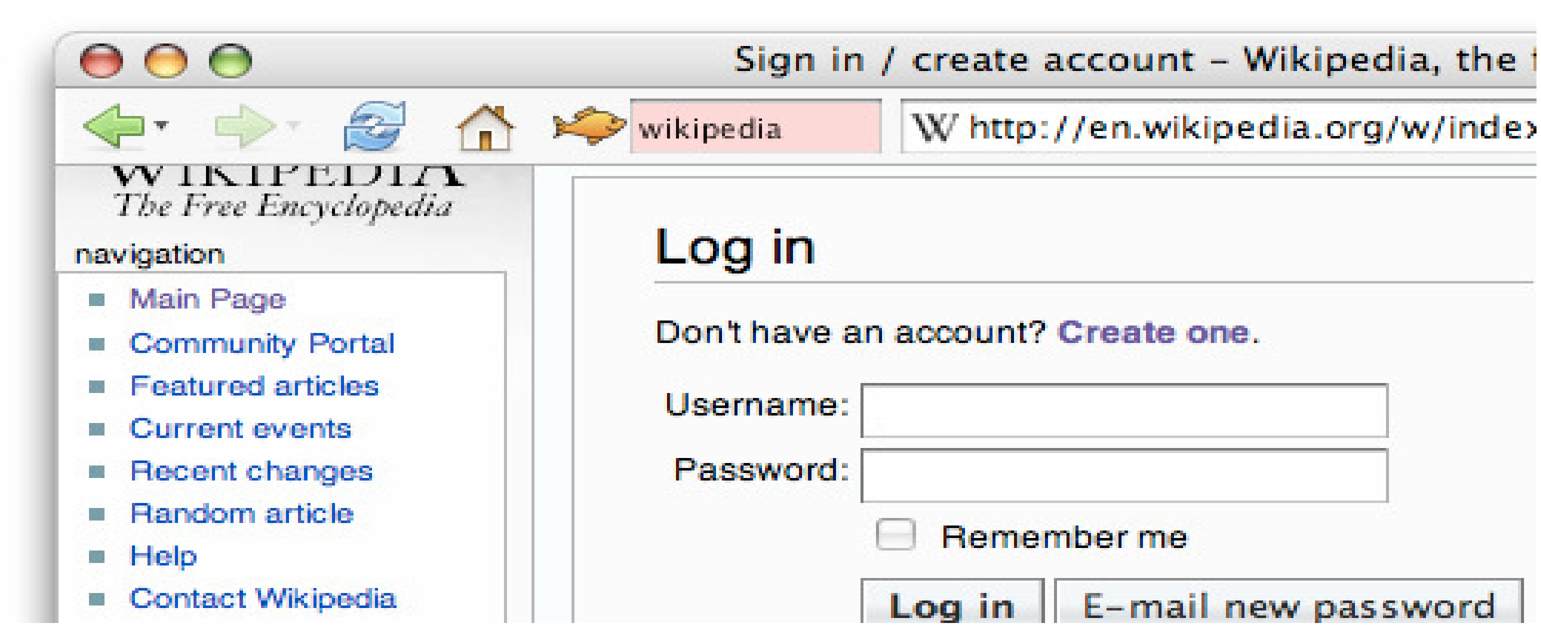
site password =
H_{k2} (site name : master password : V)

Amazon.com

Two levels of iterated hash computations

- Focuses on strengthening weak (low-entropy) passwords

Passpet



Sign in / create account - Wikipedia, the f

wikipedia

http://en.wikipedia.org/w/index

Log in

Don't have an account? Create one.

Username:

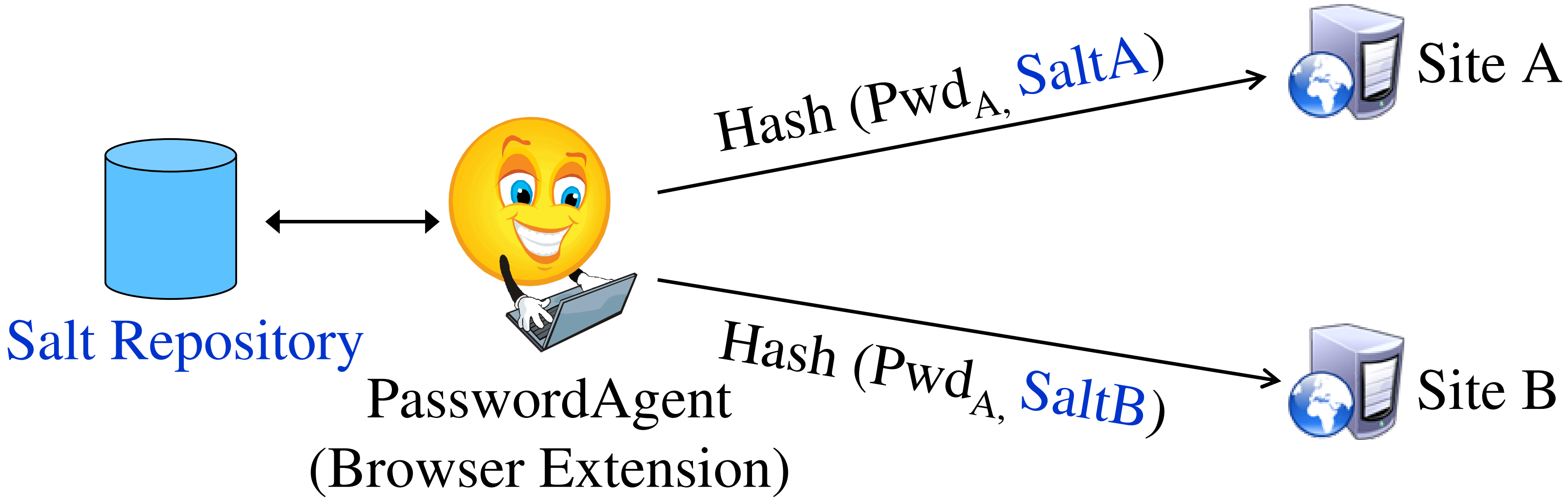
Password:

□ Remember me

(http://passpet.org)

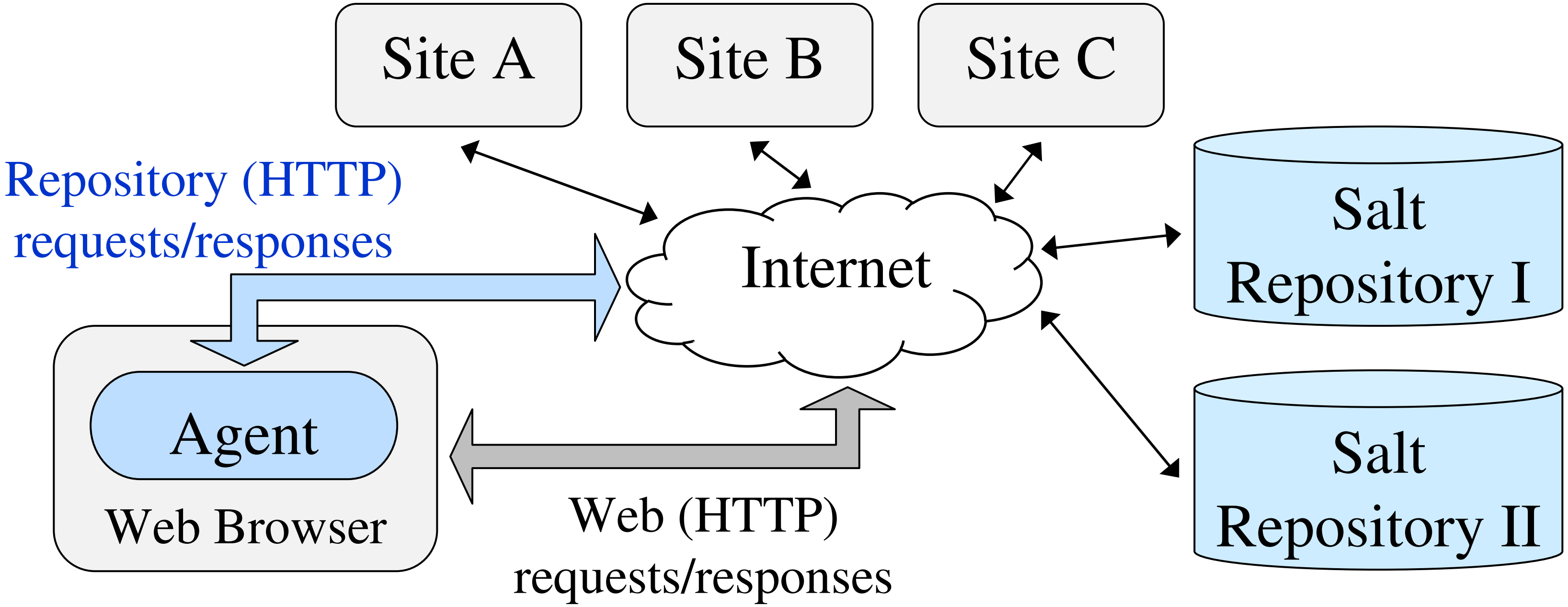
- Built upon Password Multiplier and Petname Tool
- Focuses on anti-phishing

PasswordAgent Overview



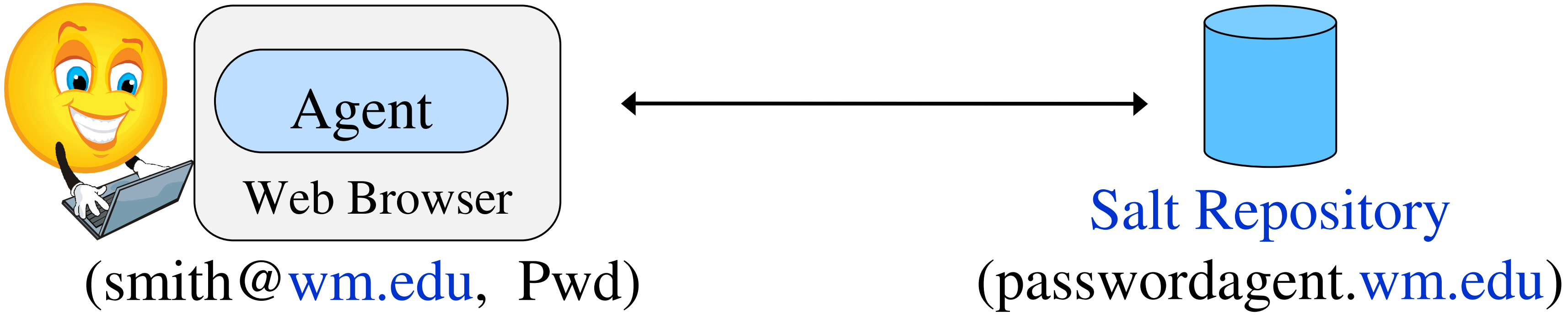
- Built upon PwdHash, introducing a salt repository
- Focuses on strengthening weak passwords, anti-phishing

PasswordAgent Architecture



- Multiple salt repositories can be used, can be switched

Installation and Setup



Agent

Web Browser

(smith@wm.edu, Pwd)

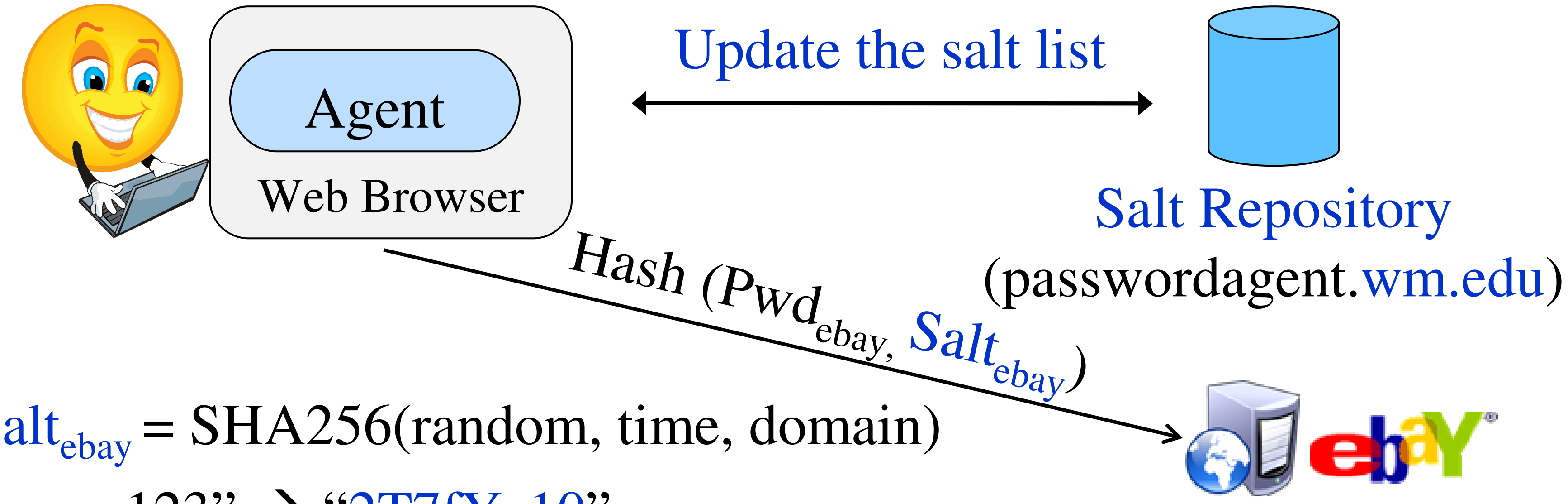
Salt Repository

(passwordagent.wm.edu)

1. Download and install the Agent
2. Registers an account (username@domain, Pwd)

Agent can easily locate the salt repository.

Website Registration



Agent

Web Browser

Update the salt list

Salt Repository

(passwordagent.wm.edu)

Hash (Pwd$_{ebay}$, Salt$_{ebay}$)

Salt$_{ebay}$ = SHA256(random, time, domain)

“susan123” → “2T7fYe10”

- Use the hashed password as the site password
- Send the encrypted salt to salt repository

User Flow in a Login Process

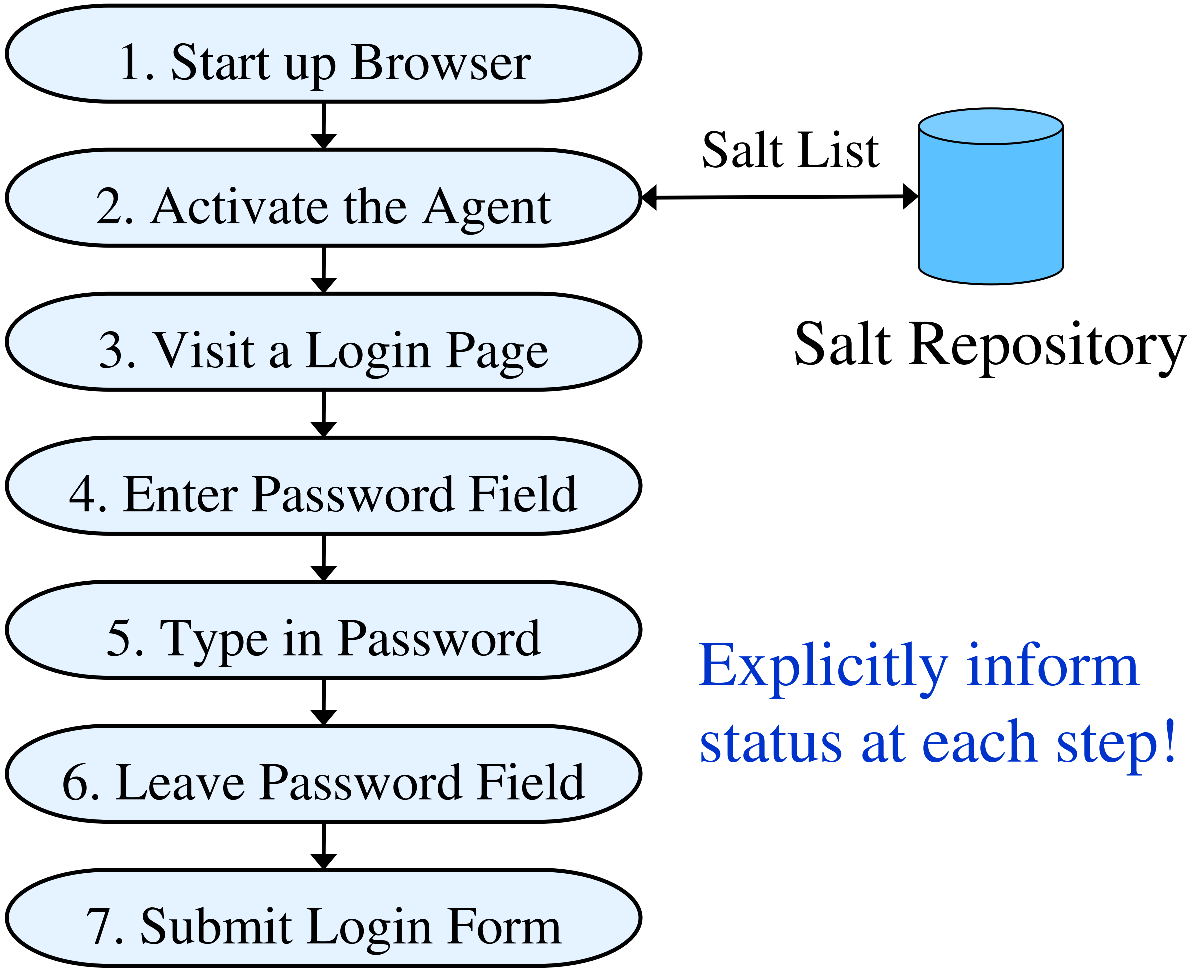
Corresponding Agent Operations

Retrieve/decrypt salt list

Lookup salt, update status

Intercept password keystrokes

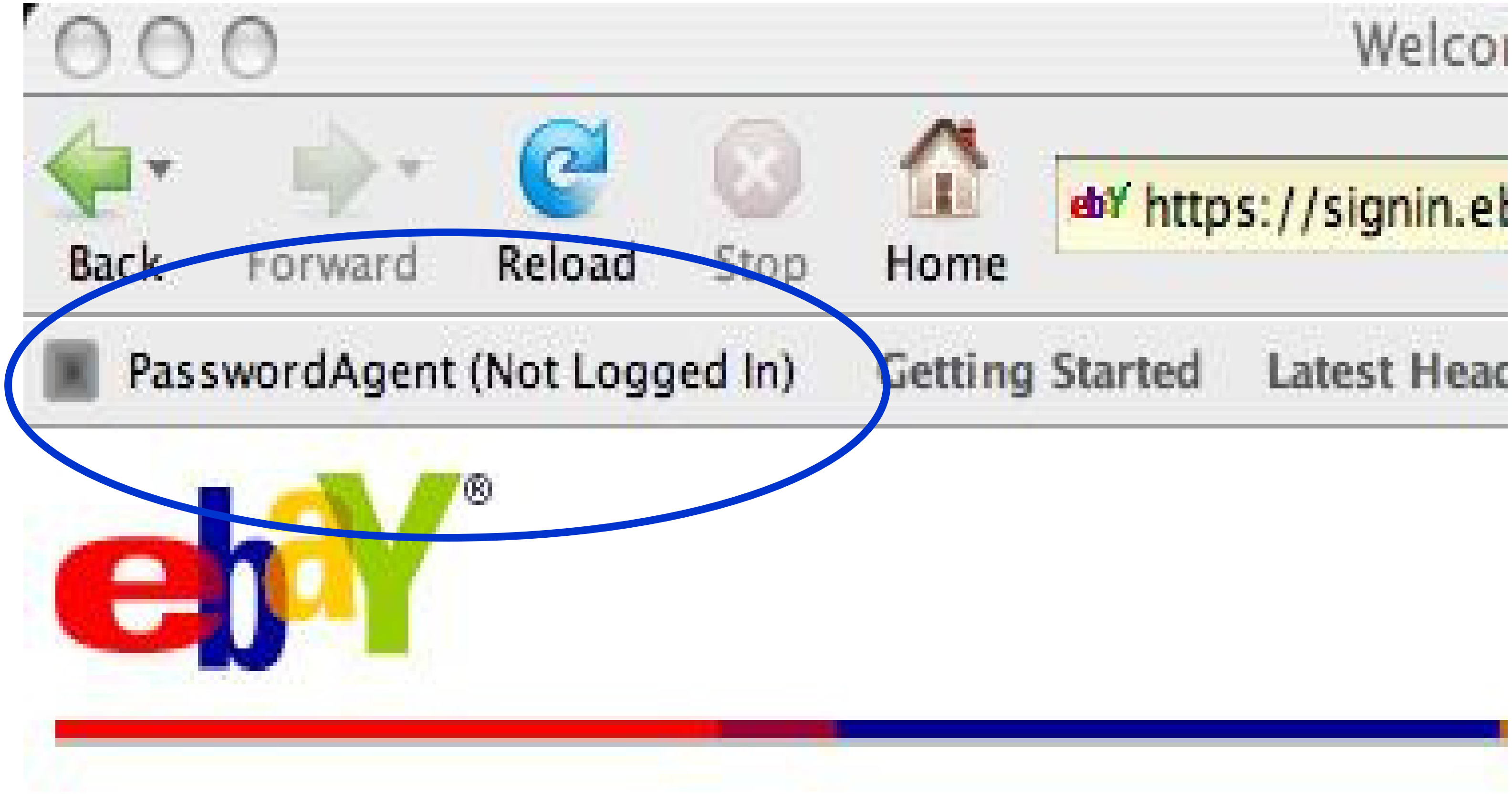
Hash, generate site password



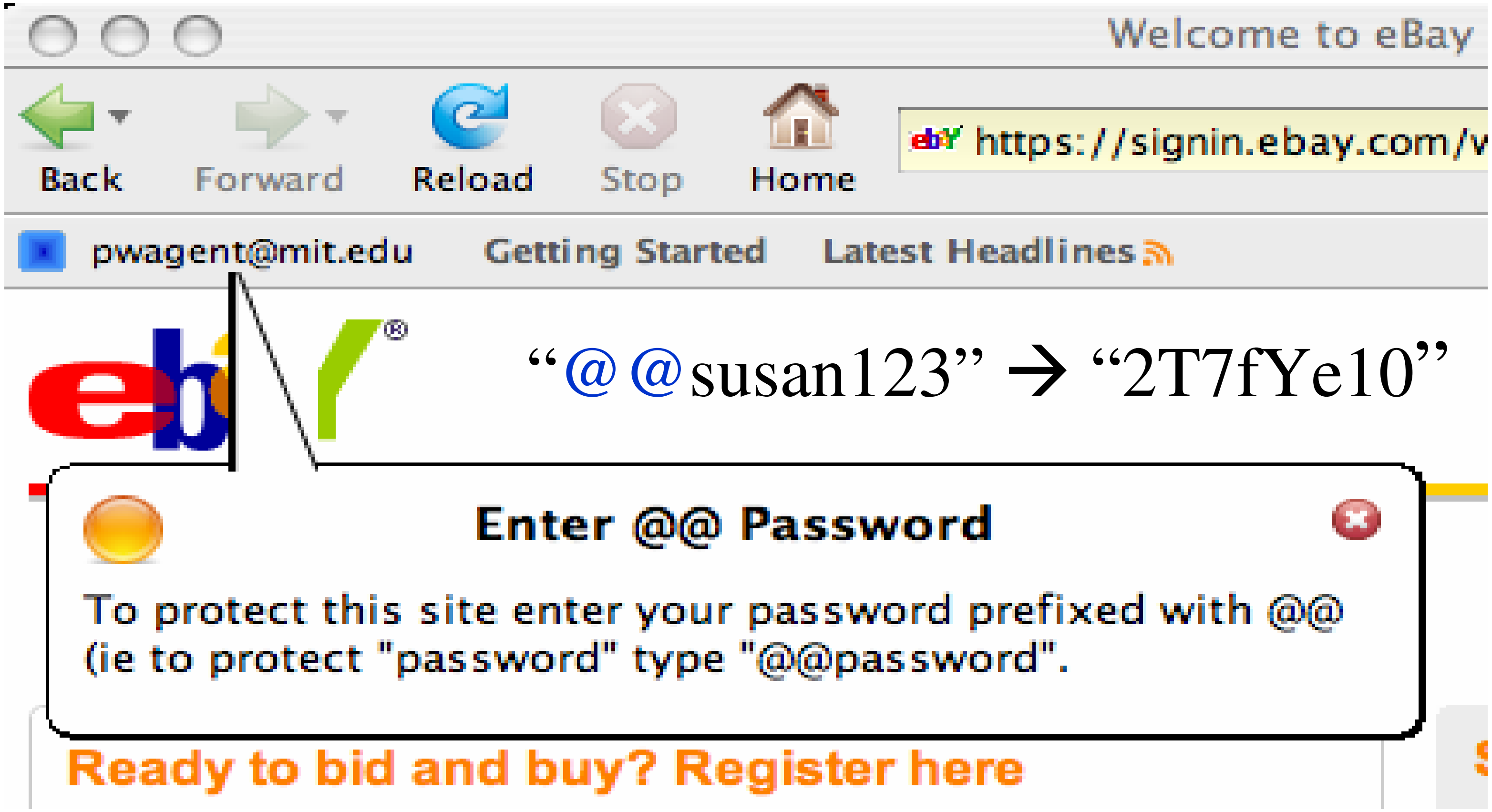
1. Start up Browser
2. Activate the Agent
3. Visit a Login Page
4. Enter Password Field
5. Type in Password
6. Leave Password Field
7. Submit Login Form

Explicitly inform status at each step!

Whether PasswordAgent is Activated?



On a Protected Website

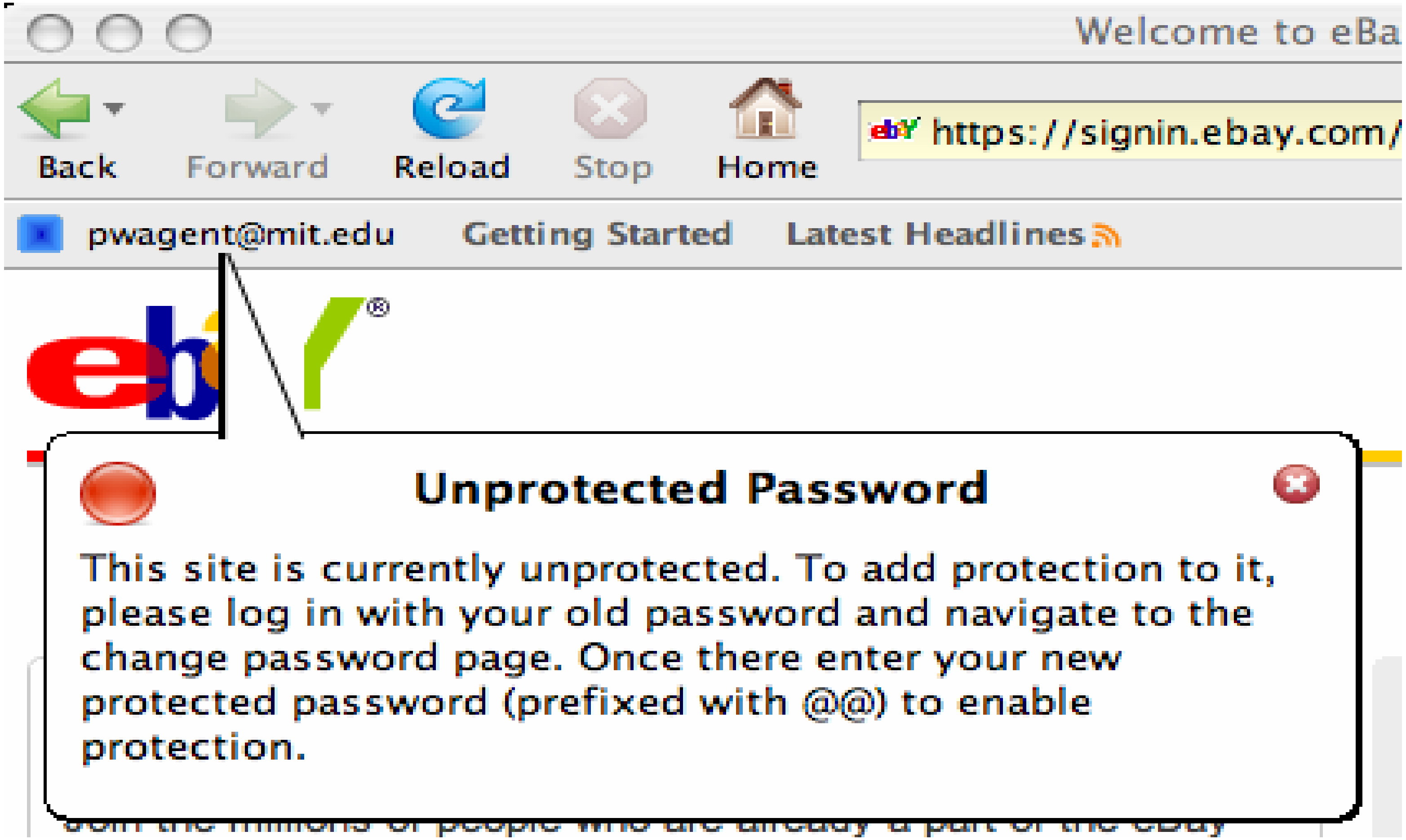


"@@susan123" → "2T7fYe10"

Enter @@ Password

To protect this site enter your password prefixed with @@ (ie to protect "password" type "@@password".

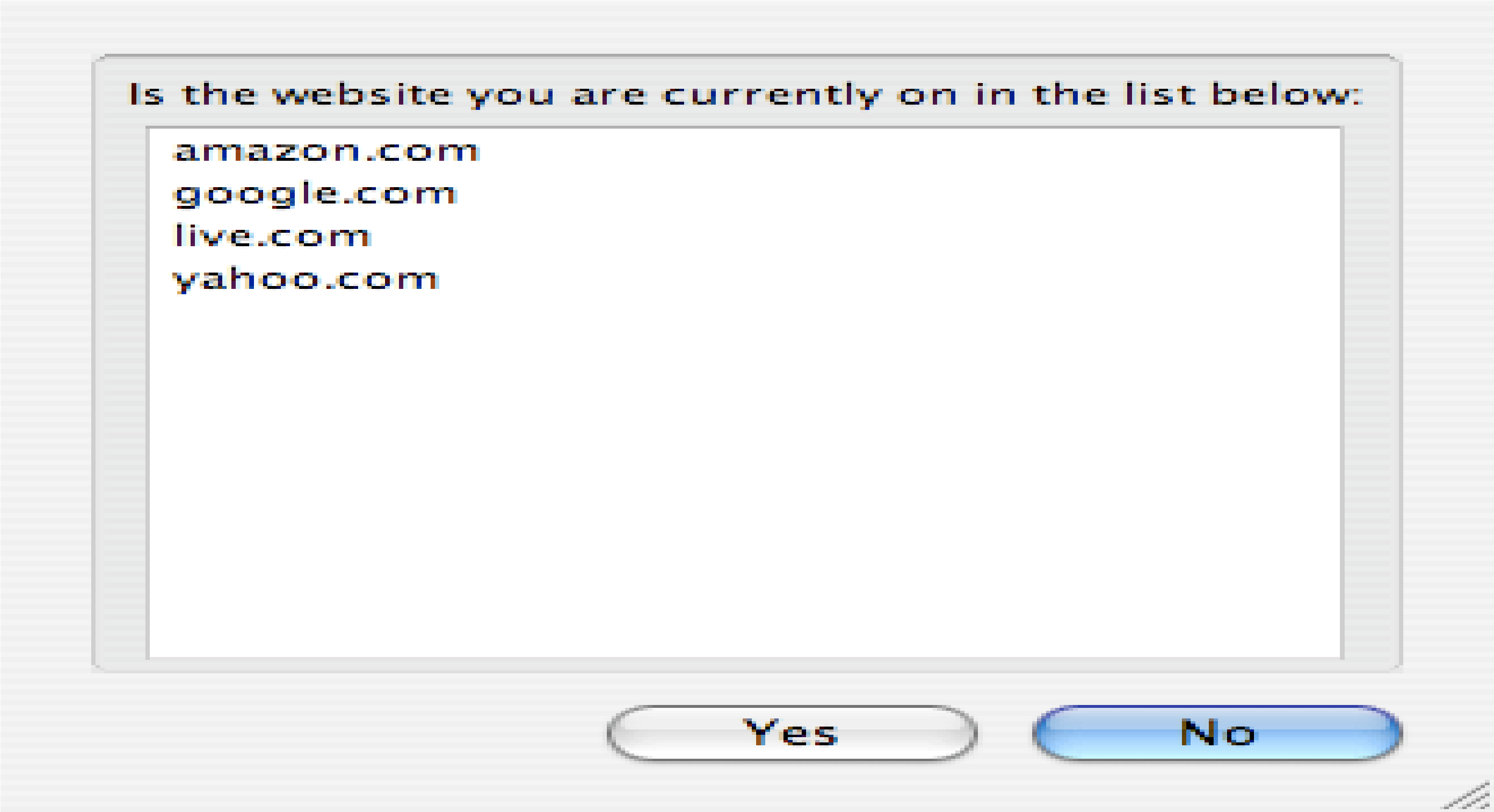
On an Unprotected Website



Unprotected Password

This site is currently unprotected. To add protection to it, please log in with your old password and navigate to the change password page. Once there enter your new protected password (prefixed with @@) to enable protection.

List of The Protected Websites



Is the website you are currently on in the list below:

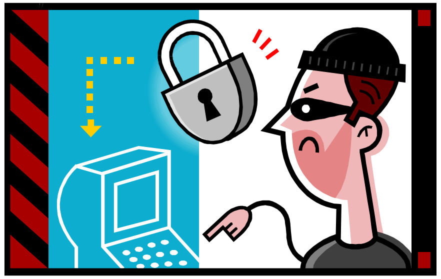
amazon.com
google.com
live.com
yahoo.com

Implementation

- Agent is a Firefox extension
  - Based on PwdHash
  - JavaScript and XUL (XML User Interface Language)

- Salt Repository is a Java Servlet
  - Hosted on an HTTPs Web server

Evaluation



Security Analysis

Usability Study

Compromised Master Password

- PasswordAgent can still protect site passwords
  - Even with stolen agent password and revealed salt list
- PwdHash does not have master passwords
- Password Multiplier and Passpet are vulnerable
  - Once the master password is compromised

Compromised Plain-text Password

- PasswordAgent can still protect a site password
  - As long as the salt is not revealed
- PwdHash cannot protect
  - Salt is known, thus site password is known
- Password Multiplier and Passpet do not have site-specific plain-text passwords

Compromised Site Password

- PasswordAgent can well protect plain-text passwords
  - Due to the large random salts
- PwdHash can protect
  - But the salt is still weak
- Password Multiplier and Passpet can well protect
  - Due to two levels of iterated hash computations

Phishing Protection

- Basic phishing protection
  - PwdHash, Password Multiplier, Passpet, PasswordAgent
- Advanced phishing protection
  - Passpet uses petname toolbar
  - PasswordAgent uses notification bubble and dialog box

Usability Study

- Twenty-eight participants (age from17 to 63)
- Each participant used PwdHash and PasswordAgent
- Five tasks
  - Migrate an unprotected account
  - Login with a protected account
  - Update the password of a protected account
  - Login with an updated password of a protected account
  - Login from another computer

Study Results

- PasswordAgent achieves higher success rates

- Comparable ratings
  - Perceived Security
  - Perceived Comfort
  - Perceived Ease of Use
  - Perceived Necessity and Acceptance

Limitations

- Vulnerable to malware such as keyloggers
- Dependence on the Salt Repository
  - Multiple synchronized repositories may help
- Usability limitations
  - Using “@@” to trigger the protection
  - Dependence on the Agent password

Summary

- A new password hashing system
- Salt Repository plus Agent browser extension
- A prototype implementation
- Security analysis and usability study
- Enhanced online password protection

Thank You!