Secure Remote Password (SRP) Authentication

Tom Wu
Stanford University
tjw@cs.Stanford.EDU
Authentication in General

- What you are
  - Fingerprints, retinal scans, voiceprints
- What you have
  - Token cards, smart cards
- What you know
  - Passwords, PINs
Password Authentication

- Concentrates on “what you know”
- No long-term client-side storage

Advantages
- Convenience and portability
- Low cost

Disadvantages
- People pick “bad” passwords
- Most password methods are weak
Problems and Issues

- Dictionary attacks
- Plaintext-equivalence
- Forward secrecy
Dictionary Attacks

- An off-line, brute force guessing attack conducted by an attacker on the network
- Attacker usually has a “dictionary” of commonly-used passwords to try
- People pick easily remembered passwords
- “Easy-to-remember” is also “easy-to-guess”
- Can be either passive or active
Passwords in the Real World

- Entropy is less than most people think
- Dictionary words, e.g. “pudding”, “plan9”
  - Entropy: 20 bits or less
- Word pairs or phrases, e.g. “hate2die”
  - Represents average password quality
  - Entropy: around 30 bits
- Random printable text, e.g. “nDz2\u>O”
  - Entropy: slightly over 50 bits
Plaintext-equivalence

- Any piece of data that can be used in place of the actual password is “plaintext-equivalent”
- Applies to:
  - Password databases and files
  - Authentication servers (Kerberos KDC)
- One compromise brings entire system down
## Forward Secrecy

- Prevents one compromise from causing further damage

<table>
<thead>
<tr>
<th>Compromising</th>
<th>Should Not Compromise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current password</td>
<td>Future passwords</td>
</tr>
<tr>
<td>Old password</td>
<td>Current password</td>
</tr>
<tr>
<td>Current password</td>
<td>Current or past session keys</td>
</tr>
<tr>
<td>Current session key</td>
<td>Current password</td>
</tr>
</tbody>
</table>
In The Beginning...

- Plaintext passwords
  - e.g. unauthenticated Telnet, rlogin, ftp
  - Still most common method in use
- “Encoded” passwords
  - e.g. HTTP Basic authentication
- Using password to encrypt verifiable text
  - e.g. Kerberos
  - vulnerable to dictionary attack
More Weak Authentication

- **Challenge-Response authentication**
  - e.g. S/Key, OPIE, CRAM
  - User receives C, responds with f(C, P)
  - Susceptible to passive dictionary attack

- **“Public-Key-Assisted” Login**
  - e.g. stel, SRA Telnet
  - Uses plain Diffie-Hellman or ephemeral RSA
  - Susceptible to active attacks
Augmenting Weak Methods

◆ Iterated hashing
  – Increases amount of time required for attack
  – Also slows down legitimate authentication
  – Maximum improvement is less than 10 bits

◆ Computer-generated passwords
  – Have higher entropy
  – Easy to forget, and are more likely to be written down (a security problem in itself)
The Empire Strikes Back

- 20-25 bits can be attacked easily today
- Even *one* successful password guess (weakest link in chain) lets an intruder in
- Attacks are easily parallelizable
- Moore’s law constantly erodes security of weak methods
  - Lose 2 bits of strength every three years
Better Weapons

- **EKE**
  - Bellovin & Merritt: 1992

- “Secret public-key”
  - Gong, Lomas, Needham, Saltzer: 1993

- **SPEKE**
  - Jablon: 1996

- **OKE**
  - Lucks: 1997
Advantages of Strong Methods

- Attacker must solve at least one “hard” public-key problem first
- Key exchange after successful login
- Some also offer:
  - Resistance to active attacks
  - Forward secrecy
- But these are still plaintext-equivalent...
Return of the Jedi

- A-EKE
  - Bellovin & Merritt: 1994
- B-SPEKE
  - Jablon: 1997
- SRP
  - Wu: 1997
- All are verifier-based
History of SRP

- Originally designed to handle authentication between Java applet and Java-based server at Stanford
- Widespread applicability and interest led to development of software suite
- Discussions on sci.crypt led to final version, sometimes called SRP-3
The SRP Protocol

\[ m = \text{large safe prime (} 2q+1, \ q \text{ prime}) \]
\[ g = \text{primitive root mod } m \]
\[ P = \text{plaintext password} \]
Carol knows the password, \( x = H(P) \)
Steve knows the verifier, \( v = g^x \mod m \)

Carol (the client)
Generates random \( a \)
\[ A = g^a \mod m \]
\[ K = (B - g^x)^a + ux \mod m \]

Steve (the server)
Generates random \( b, u \)
\[ B = v + g^b \mod m, u \]
\[ K = (Av^u)^b \mod m \]

Each side then proves it knows \( K \)
Security Analysis

- $g$ must be primitive root to avoid leaking information about $v$
- Carol must be first to issue proof of $K$
- $u$ must not be revealed before $A$
- Random numbers $a, b$ must be discarded when protocol finishes
- Check for invalid inputs, e.g. $A, B == 0$
Protocol Families

- A- and B- protocols sometimes called “extended” methods
  - Two key-exchange rounds
  - A-EKE: extra digital signature
  - B-SPEKE: extra El Gamal key exchange

- SRP is part of a third family (AKE)
  - Password and verifier are integrated into a single key-exchange round
### How They Stack Up

<table>
<thead>
<tr>
<th></th>
<th>Cleartext Password</th>
<th>Challenge-Response</th>
<th>Strong Password (SPEKE)</th>
<th>Verifier-based (SRP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does not reveal password</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Resists dictionary attack</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Not plaintext-equivalent</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Provides forward secrecy</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Some Scary Trends

- Inexpensive PCs can exceed 10K crypts/sec
- No Moore’s Law for human memory
- Increased size of Internet means more access to free CPU cycles
- Even forcing “good” password choices only delays inevitable
- “Bad guys” have caught up and taken the lead (CERT Advisory 94:01)...
Strong vs. Weak Methods

Effort needed to crack an average password

Cost of attack (PC * hours)

1E+12
1E+9
1E+6
1E+3
1E+0
1E-3


SRP
Iterated Hash
C-R
(safe)
Impact of Strong Authentication

- Best known algorithms for compromising SRP (discrete log) not easily parallelizable
- Places off-line attack against even short passwords out of reach
- “Infinitely” stronger than weak methods
- Designed as drop-in replacements
- Low cost and socially acceptable
Applications

- Remote login/access (Telnet, FTP)
- E-mail (POP, IMAP)
- World Wide Web
- Firewalls
- Network computers
- Any situation where an actual person needs to be authenticated
The SRP Project

- Freeware API library in both C and Java
- Telnet and FTP for Unix and Windows
- Distributed architecture being developed
- Public collaboration w/Internet community
- Drafts submitted to IETF, IEEE P1363
- Help always appreciated
  - Join the mailing list
Looking Ahead

- Increase awareness of “broken” systems
  - Kerberos V4, V5, Windows NT, S/Key, ...

- Fix these systems where possible
  - Jaspan, 1996 describes Kerberos fix

- Strong authentication mechanisms have become “Best Current Practice”
  - No excuse to use broken methods anymore
  - Good guys are winning again...
For More Information

- SRP Web Site
  - http://srp.stanford.edu/srp/
  - Contains links to download source code and information about mailing list

- E-mail
  - tjw@cs.Stanford.EDU