Authentication Methods

Passwords

PRESENTED BY:

SHREYAS PRABHAKAR

Text-based Passwords

- Passwords form the foundation of security policy for broad spectrum of online service.
- Simple and most commonly used authentication mechanism.
- Does not require any special hardware
- Does not require user to carry anything
- Easy for the end user to input
- Easy to deploy and incorporate with existing services.
- Lack of need to memorize or store complex cryptographic keys

Properties of Human Memory

- Human memory for sequence of items is *temporally limited*, with a short-term capacity of around seven.
- When humans remember a sequence of items, they cannot be drawn from an arbitrary and unfamiliar range, but *must be familiar "chunks"*, such as words or familiar symbols.
- Human memory *thrives on redundancy* – better at remembering information that can be encoded in multiple ways

Folk beliefs about passwords

- Users have difficulty remembering random passwords.
- Passwords based on mnemonic phrases are harder for an attacker to guess than naively selected passwords are.
- Random passwords are better than those based on mnemonic phases
- Passwords based on mnemonic phrases are harder to remember than naively selected passwords.
- By educating users to use random passwords or mnemonic passwords, we can gain a significant improvement in security.
- Any password that contains letters, digits, and symbols is secure.

Password Policy

- Series of rules typically instituted by administrators and organizations
- Consensus in literature – Properly written password policy can provide an organization with increased security.
- Effects of password policies are still unclear –
  - Difficult to determine the practical password space.
  - Some Password policies, resulting in stronger passwords –
    - Make passwords difficult to remember and type
    - Users end up writing or reusing or sharing passwords
    - Increase in Frequency of forgotten passwords - increases help-desk workload and IT-support cost
    - Discontent among users
    - Generally diminished productivity

Password Constraints across different websites

- Char5: Minimum 5 characters.
- Char6: Minimum 6 characters.
- Char8: Minimum 8 characters.
- Char6LU: Minimum 6 characters containing at least one lowercase letter and one uppercase letter.
- Char8DSU: Minimum 8 characters containing at least one number, symbol, or uppercase letter.
- Char8LDS: Minimum 8 characters, with at least one letter (either uppercase and/or lowercase) and at least one number and/or symbol.
- Char6D: Minimum 6 characters containing at least one letter and one number.

Password Constraints (contd..)

- Char6-12: Contains 6-12 characters. Characters can be letters, digits or even symbols.
- Strong1: Contains 7-32 characters with at least one letter and one number. Cannot include special characters (&, %, *, etc.). Cannot be the same as user ID and cannot be the same as any of the last five passwords used.
- Strong2: Contains 8-20 characters with at least one letter and one number. Cannot include any spaces or the following special characters $, <, >, &, ^, !, [,]. Cannot be the same as user ID. Password is case-sensitive.
- Strong3: Contains 8-20 characters with at least one letter and one number. May include the following characters: %,&, , ?, #, =, -. Cannot have any spaces and will not be case sensitive. Must be different from user ID.

Example:

I. Passwords must not contain the user's entire name/user ID.

II. At least n characters (usually n 6).

III. Passwords must contain characters from two or more of the following four categories:

1: Uppercase characters (A through Z)

2: Lowercase characters (a through z)

3: Base 10 digits (0 through 9)

4: Non-alphanumeric ASCII characters:

~!@%^&* -+=—(){}[]:;”‘’<>,.?$\

CMU Policy Change – Dec 2009

- At least 8 characters.
- Include at least one upper-case letter, one lowercase letter, one digit, and one symbol.
- Subject to dictionary check
- Passwords containing four or more occurrences of the same character would also be rejected.

Affect of Password Policies on Users

- Although a password policy that allows weak passwords can lead to system compromise, an overly strong policy can lead to users writing down passwords and thereby increase system vulnerability.
- Although users were aware of security concerns
  - Rarely changed their passwords
  - Password policies did not account for sensitivity variations in resources they protect.
- Common password length was 6 characters
- Automated cracking tools were less successful against mnemonic passwords than against control passwords
- On average, each person used seven passwords across 25 different websites.

Human Algorithm or Strategies while creating passwords

- Create passwords based on the security value of the account
- Order of thought process : series of letters, then digit, then symbol
- Password is built from left to right as participants think of elements
- Use site-specific information or objects around them
- Name of participant, immediate family member, or pet, or date or geographic location
- Reuse same password, or with modification
- Use longer words and add a digit at the end

Password Reuse

- Typical Internet user is estimated to have 25 distinct online accounts.
- Users often reuse passwords across accounts on different online services.
- Vulnerable to cross-site password attacks
- Password reuse cannot be prevented by traditional composition policies or meters, as these tools only see passwords at a single site.

Password Similarity metrics

- **Distance-like functions:** These functions compute the distance between two strings by first mapping each string into a point in a multidimensional space and then computing the distance between those points.
- **Edit-distance like functions**: These functions determine the number of edit operations (insertion, deletion, replacement, transposition) required to transform one string into another.
- **Token-based distance functions**: These functions first split the strings in smaller tokens (i.e., bigrams) and then compute the similarity between them.
- **Alignment-like functions**: This set of functions provide similarity scores that reflect the largest alignment or subsequence between a pair of strings

Commonly used Transformation Rules

- Number at the end of password string
- Insert Uppercase at the beginning
- Inserting a symbol at either the middle or the end
- Have a base word and append a variation of website name
- Character Substitution
- Replace birthday with favourite 4-digit sequence
- Insert ‘@’ at the beginning (reminded of twitter)
- Use emoticon at the end

Guessing Algorithm

Input: Input password *a* and target password *b*

Intermediate result: Candidate password *a**

Output : Cracked or not cracked.

Check (a*,b) : if a* = b **return** cracked

If a contains any sequential pattern **then**

    sequential transformation(a) -> a*

    Check (a*,b)

End if

If len(a) > 6 **then**

    a* <- deletion(a)

    Check (a*,b)

End if

If len(a) < 10 **then**

    a* <- insertion(a)

    Check (a*,b)

End if

Capitalization(a) -> a*

Check (a*,b)

Reverse(a) -> a*

Check (a*,b)

Leet(a) -> a*

Check (a*,b)

Substring Movement(a) -> a*

Check (a*,b)

Sub word transformation(a) -> a*

Check (a*,b)

Return not cracked

Other Guessing Algorithms

- RockYou Guesser
- Edit Distance (ED) Guesser
- John the Ripper (JTR)

Countermeasures for Cross-site Security

- Single Sign-on Technologies
- Two – factor Authentication
  - User ID and Password
  - Verification Code sent to phone
- Educating users of the importance of using substantially different passwords across sites.
- Develop a cross-site password security metric

Advice and Recommendations

- Promote secure Human Algorithms
- Choose mnemonic-based passwords.
- Size matters
- Assigning values to accounts
- Understanding Threats
- Entropy per character also matters
- Compliance is the most critical issue
- Better Data-Driven Feedback

Bad Strategies for Password Creation

- Use of Dictionary words and Birthdays
- Use of common Keyboard Patterns
- Choosing obvious phrases (Eg: “ilove*SiteName*”)
- Adding Digit or Symbol at the end
- Thinking that words that are hard to spell are secure