CSCI369 Ethical Hacking

Lecture 4-1 Passwords and Password Cracking

A/Prof Joonsang Baek

School of Computing and Information Technology

UNIVERSITY OF WOLLONGONG AUSTRALIA

Introduction to Password Cracking

- Password cracking
  - ➢A starting point for any serious exploitation.
  - ➢An effective way to gain access to a system.
  - ➢Many institutions educate their members to select strong passwords, but it is hard to expect that everyone follows the rule.
    → Almost always weak passwords are used.

UNIVERSITY OF WOLLONGONG AUSTRALIA

Password Cracking Techniques

- Dictionary attack
  - Uses a list of words which can possibly used as passwords.
  - Password cracking software usually has *pre-loaded* lists of words or allows users to load their own list.
  - The words on the list can accelerate the cracking process and time.
  - These lists can also be downloaded from many websites (for free); some may contain millions of items.

Password Cracking Techniques

- Brute-force attack
  - ➢Every possible combination of *characters* is attempted until the correct one is discovered
  - ➢Although this attack has potential to be successful eventually, many modern systems employ techniques such as “account lockouts” and “bad login counts (threshold)” to prevent this attack.
    - ✓Usually, threshold values are 3 to 5 (attempts).
    - ✓Once the limit has been reached, the account will be locked out and will require an administrator to reset the password.

Password Cracking Techniques

- Hybrid attack
  - ➢This attack uses the dictionary attack as a basis but adopts some techniques of brute-force attack as part of the process.
    - ✓For example, it can attempt words in the dictionary but add numbers and/special characters in a brute-force way.

Password Cracking Techniques

- Passive online attack
  - ➢This attack achieve its goal by tapping into the network and using a sniffer or an MITM tool to observe the traffic that may contain passwords.
  - ➢Sniffer
    - ✓A tool such as Wireshark to passively observe network traffic to capture passwords.
  - ➢MITM tool
    - ✓A tool that launches man-in-the-middle attacks (such as Bettercap) to capture passwords.

UNIVERSITY OF WOLLONGONG AUSTRALIA

Password Cracking Techniques

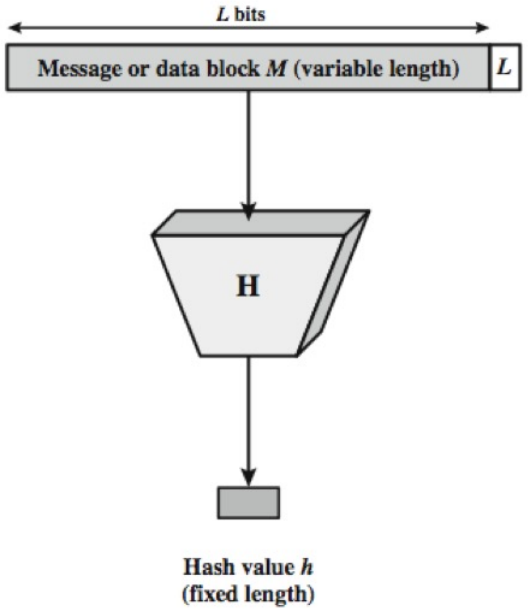
- Active online attacks
  - ➢This attack takes more aggressive approaches to extract passwords by engaging with the targets.
  - ➢The attack is based on malware, which can potentially obtain some other information as well as passwords during the process.
  - ➢The malware includes Trojans, spyware, and *key loggers*.
    - ✓Keyboard sniffing or key logger intercepts the password as a user enters it; this attack can be executed using hardware- or software-based mechanisms.

Password Cracking Techniques

- Offline attacks
  - ➢Because passwords cannot be stored in a clear format, a hash function should be applied to passwords and the resulting hash values (digests) are stored in the system.
    - ✓This is to protect the passwords from being exposed as the hash function has the one-way property.
  - ➢Offline attacks are performed using the hash values the attacker has obtained.

Password Cracking Techniques

- Basic facts about hash function
  - ➢A hash function (in cryptography) is a cryptographic algorithm that takes an arbitrary input returns a fixed size output
  - ➢The size of hash value (h) depends on the type of hash algorithms
    - ✓SHA-1: 160 bits (nearly broken)
    - ✓MD5: 128 bits (broken)
    - ✓SHA-2: 224, 256, 384 and 512 bits
    - ✓SHA-3: 224, 256, 384 and 512 bits
  - ➢The hash function is one-way and collision resistant.



UNIVERSITY OF WOLLONGONG AUSTRALIA

Password Cracking Techniques

- Rainbow table
  - ➢Well known pre-computed table for reversing hash.
  - ➢The basic idea is to search p such that $h = \mathrm{H}(p)$ using a huge number of pre-computed (p, h)'s.
    - ✓Table look-up is much faster than computing hash values one by one.
    - ✓This technique is called time/memory trade off.
  - ➢The size of this table is usually big as shown in the following website: http://project-rainbowcrack.com/table.htm
    - ✓Nevertheless, the size is manageable as the storage is getting cheaper and cheaper recently.

Password Cracking Techniques

- Use of *salt* for password hashing
  - Computing $h = \mathrm{H}(p||salt)$ instead of $h=\mathrm{H}(p)$, where salt is a (reasonably large) random number
  - Note that *salt* should be public (non-secret) to check the hash value
- Two main reasons why the use of salt is good
  - The possibility of collision (m and m' such that $\mathrm{H}(m) = \mathrm{H}(m')$) can be reduced
  - $p||salt$ enlarges the input space of the hash function, so the size of rainbow table has to be much bigger, which makes the rainbow table attack impossible.

UNIVERSITY OF WOLLONGONG AUSTRALIA

Password Cracking Techniques

- Where can we find salt in Linux password
  - ➢Look at `/etc/shadow`
  - ➢For example, here is one of the entries for a user in the Linux system:
    `$y$j9T$TcUtaKk36hgKNxMVGist5.$m7q5zSMErBMunR.kEk3XcXMzVX6ojE9vVJgeU3m6um1`
    - ✓"y" means hash algorithm name – yescrypt hash in the current Kali Linux. In other Linux, 6 means SHA-512, which is common recently.
    - ✓ "`j9T$TcUtaKk36hgKNxMVGist5.`" is the salt
    - ✓"`m7q5zSMErBMunR.kEk3XcXMzVX6ojE9vVJgeU3m6um1`" is Hash(password||salt)

Securing Passwords with Hash

- Secure password hashing guidelines
  - ➢Do not use MD5 or SHA1 for storing passwords.
  - ➢Use SHA2 or SHA3, preferably.
    - ✓SHA2 output options: 224, 256, 384 and 512 bits
    - ✓SHA3 output options: 224, 256, 384 and 512 bits (The same as SHA2)
  - ➢Use salt to mitigate against password brute-force attacks.

Password Cracking Techniques

- Non-technical attacks: Also known as non-electronic attack, this does not use electronic means to crack passwords.
  - ➢ *Default passwords*
    - ✓ Using default passwords to gain access is seemingly trivial but is surprisingly very effective.
    - ✓ Default passwords are those set by the manufacturer or software developer.
    - ✓ In fact, these passwords are supposed to be changed by the customers when they set up purchased devices or software.
    - ✓ The problem is that quite a large number of users leave the default setting intact.
    - ✓ Some sites have collected default passwords: http://www.defaultpassword.com/

Password Cracking Techniques

- Social engineering
  - ✓Like using default passwords, this attack can be effective even if no hardware/software tool is used.
  - ✓Blackmail or conciliate or deceive users to give up their passwords through non-technical means.

Password Cracking Tools

- Hydra
  - Online brute-force password attack tool.
  - One of the most popular password cracking tools.
  - Makes use of numerous protocols including ftp, http, ssh, smtp, POP3, mysql and etc.
  - Supports multiple connections (i.e. parallel attacks).
  - A list of candidate passwords should be provided. (It can be obtained externally.)
  - Basic command line syntax is

```
hydra [options] <target IP> <protocol>
```

Password Cracking Tools

- Hydra example

```
hydra –t 16 –l victim -P /usr/share/john/password.lst –vV
<MetasploitableIP> ftp
```

  - `-t`: tasks meaning the number of connects in parallel per target (default: 16)
  - `-l`: login with a given login name, in the above example, "`victim`"
  - ✓ With –L, we can provide a dictionary file for username
  - `-P` file: load several passwords saved in the file (`password.lst`)
  - `-v`: verbose
  - `-V`: shows every password being tried

Password Cracking Tools

- John-the-Ripper
  - Different from Hydra in that it basically finds the target's password using offline attack.
  - It usually uses a password list but is also capable of performing brute-force attack.
  - In fact, it is to find a pre-image of hash values: The pre-image is the password the attacker is seeking.

Password Cracking Tools

- The attack consists of two steps:
  1. Combine `/etc/passwd` and `/etc/shadow` files. (This process is called "unshadowing")
     - ✓ `/etc/passwd` stores a list of registered users in the system and `/etc/shadow` stores the hashes of the passwords.
     - ✓ Note that the earlier version of UNIX system's passwd file stored the hashes of the passwords.
  2. Performing a dictionary attack against the unshadowed file using a word list to find a password.

Password Cracking Tools

- John-the-Ripper example
  - Combine entries of `/etc/passwd` and `/etc/shadow` by unshadowing: `unshadow /etc/passwd /etc/shadow > ./victims_pwd`
  - Run John the Ripper using the password list:

```
john --wordlist=/usr/share/john/password.lst victims_pwd
```

Dilemmas of Passwords

- Two characteristics of convenient yet strong passwords
  - Easy to remember
  - Not easily guessed or broken
- In reality
  - It is not easy to satisfy both.
  - Even if users are told to mix up characters (lowercase/uppercase), special characters and numbers to create *strong passwords*, the way they are created can lead an attack.

Week Passwords

- Default passwords (as supplied by the system vendor and meant to be changed at installation time): password, default, admin, guest, etc. Lists of default passwords are widely available on the internet.
- Dictionary words: chameleon, mustang, sandbags, bunnyhop!, IntenseCrabtree, etc., including words in non-English dictionaries.
- Words with numbers appended: password1, deer2000, john1234, etc., can be easily tested automatically with little lost time.
- Words with simple obfuscation: p@ssw0rd, @dm1n, l33th4x0r, g0ldf1sh, etc., can be tested automatically with little additional effort.

UNIVERSITY OF WOLLONGONG AUSTRALIA

Week Passwords

- Doubled words: crabcrab, stopstop, treetree, passpass, etc.
- Common sequences from a keyboard row: qwerty, 12345, asdfgh, fred, etc.
- Numeric sequences based on well known numbers such as 911 (9-1-1, 9/11), 314159... (pi), 27182... (e), 112 (1-1-2), etc.
- Identifiers: jsmith123, 1/1/1980, 333–1234, one's username, etc.
- Weak passwords in non-English languages, such as contraseña (Spanish) and ji32k7au4a83 (“my password” encoding from Chinese)

UNIVERSITY OF WOLLONGONG AUSTRALIA

Good Passwords

- Simply, good passwords are random passwords consist of a string of symbols of specified length taken from some set of symbols using a random selection process.
- Each symbol is equally likely to be selected.
- The symbols can be individual characters from a character set (e.g., the ASCII character set), syllables designed to form pronounceable passwords, or even words from a word list (thus forming a passphrase).

Good Passwords

- Measuring a password strength
  - A measure of the effectiveness of a password against guessing or brute-force attack.
  - password_strength=*f*(length, complexity, unpredictability)
- Password creation
  - By machine
  - By human
    - ✓Humans tend to follow patterns.
    - ✓Prone to attack.

Random Passwords

- Most modern operating systems use cryptographically strong random number generators to generate passwords.
- Random password programs often ensure that the resulting password complies with a local password policy.
  - For example, a policy might say that a password should consist of a mix of letters, numbers and special characters.

Random Passwords

- Information entropy H
  - Measures a strength of a random password.
  - The unit is “bits”.
- Formula:
  - $H = \log_2 N^L$, where L is the length of a string (password) and N is the number of possible symbols.
  - It can be easily derived that $L = H/\log_2 N$

Random Passwords

- Table: Lengths L of randomly generated passwords required to achieve a desired password entropy H for sets containing N symbols.

Desired password entropy *H*	Arabic numerals	Hexadecimal	Case insensitive Latin alphabet	Case insensitive alphanumeric	Case sensitive Latin alphabet	Case sensitive alphanumeric
8 bits (1 byte)	3	2	2	2	2	2
32 bits (4 bytes)	10	8	7	7	6	6
40 bits (5 bytes)	13	10	9	8	8	7
64 bits (8 bytes)	20	16	14	13	12	11
80 bits (10 bytes)	25	20	18	16	15	14
96 bits (12 bytes)	29	24	21	19	17	17
128 bits (16 bytes)	39	32	28	25	23	22
160 bits (20 bytes)	49	40	35	31	29	27
192 bits (24 bytes)	58	48	41	38	34	33
224 bits (28 bytes)	68	56	48	44	40	38
256 bits (32 bytes)	78	64	55	50	45	43

UNIVERSITY OF WOLLONGONG AUSTRALIA

How Much Entropy?

- The minimum number of bits of entropy needed for a password depends on the threat model for the given application.
- According to RFC 4086 (Randomness Requirements for Security):
  - Roughly, 29-bit entropy is needed if only online attacks are expected.
  - At least, 96-bit entropy is needed for important cryptographic keys used in applications like encryption, where the password or key needs to be secure for a long period of time, and stretching is not applicable.
- Where does this estimation come from?
  - Find a password that has *n*-bit entropy = Find a random (uniform) *n*-bit key

UNIVERSITY OF WOLLONGONG AUSTRALIA

How Much Entropy?

- Calculation: To find a random key by brute-forcing
  - n: key size (in bits)
  - c: CPU speed (Usually GHz/s) → Clock cycles (cc) per second
  - Time taking for brute-force attack: $t = 2^n/c$ (Assumption: it takes one clock cycle (cc) to perform one instance of search)
  - Example

    60-bit key searched by a computer with i5 4GHz processor, i.e. $c = 4*10^9$ cc/s

    $t= 2^{60}/(4*10^9)$ s = 9 years

    60-bit key searched by a super computer with $c =2*10^{16}$ cc/s

    $t= 2^{60}/(2*10^{16})$ s = 57 seconds!

How Much Entropy?

- Then, how about other key size?
  - ➢Example

    80-bit key searched by a super computer with c =$2*10^{16}$ cc/s

    t= $2^{80}/(2*10^{16})$ s = 2 years!

    128-bit key searched by a super computer with c =$2*10^{16}$ cc/s

    t= $2^{128}/(2*10^{16})$ s = $2*2^{48}$ years! (approximately equivalent to the time since Big Bang)

    Hence n=128 is very safe

- Due to limitations from fundamental physics, there is no expectation that any digital computer (or combination) will be capable of breaking 256-bit random string via a brute-force attack.

UNIVERSITY OF WOLLONGONG AUSTRALIA

Analysing Human-Generated Passwords

- Limitation of human-generated passwords
  - ➢Insufficient entropy:
    - ✓According to one study involving half a million users, the average password entropy was estimated at 40.54 bits.
  - ➢Non-uniformity problem:
    - ✓Humans do not choose passwords uniformly at random: In one analysis of over 3 million eight-character passwords, the letter "e" was used over 1.5 million times, while the letter "f" was used only 250,000 times. The most common number used is "1", whereas the most common letters are a, e, o, and r.

Analysing Human-Generated Passwords

- Biased selection problem:
  - ✓Users rarely utilise larger character sets in forming passwords. For example, hacking results obtained from a MySpace phishing scheme in 2006 revealed 34,000 passwords, only 8.3% used mixed case, numbers and symbols.
- Predictable pattern problem:
  - ✓Even if humans are advised to choose “varied” passwords , humans are likely to choose them in predictable ways, such as capitalizing only one letter, adding one or two numbers followed by one special character.
  - ✓This predictability results in minor increase in password strength compared to random passwords (generated by machines).

Guidelines for Password Management

- Use a minimum password length of 8 or more characters if permitted.
- Include lowercase and uppercase alphabetic characters, numbers and symbols if permitted.
- Generate passwords randomly where feasible.
- Avoid using the same password twice (e.g., across multiple user accounts and/or software systems).

Guidelines for Password Management

- Avoid repetition of characters, keyboard patterns, dictionary words, letter or number sequences, usernames, relative or pet names, romantic links (current or past) and biographical information (e.g., ID numbers, ancestors' names or dates).
- Avoid using information that is or might become publicly associated with the user or the account:
  - Avoid to use birthdays, birthplace, favourite sport teams, license plate number, Medicare number, telephone numbers, student ID, addresses, pet's names, nicknames, etc.

Guidelines for Password Management

- Writing a password on a paper
  - Some argue that forcing users to remember passwords without assistance can only accommodate weak passwords, and thus poses a greater security risk.
  - But Bruce Schneier argues that since most people are good at securing their wallets or purses (called “great place”), it may be okay to write a passwords on a paper and store it in them.

Guidelines for Password Management

- Memorable password
  - According to the paper “The Memorability and Security of Passwords (IEEE Security and Privacy Magazine), Jeff Yan et al. found that passwords based on thinking of a phrase and taking the first letter of each word are memorable yet just as hard to crack as randomly generated passwords.
  - For example, “this is a good password → tiagp”

Guidelines for Password Management

- A password manager can generate and retrieve complex passwords.
- It stores generated passwords in an encrypted database or calculates them on demand.
- Types of password managers include:
  - locally installed software applications
  - online services accessed through website portals
  - locally accessed hardware devices that serve as keys
- Drawback: The loss of the master key for the password manager

Guidelines for Password Management

- Design errors
  - ➢Attempts to make passwords memorable are a frequent source of severe design errors, e.g., security questions
  - ➢The number of applications demanding a password exceeds the power of human memory → Sharing passwords across different sites would occur.
    - ✓A 2007 study by D Florencio and C Herley showed that the average user has 6.5 passwords, each shared across 3.9 different sites; types 8 passwords per day.
  - ➢Modern web browsers cache passwords but many people use the same password for *high-value logons* – e.g., the same password for the Internet banking and social media accounts.

Guidelines for Password Management

- Pervasive and persistent design errors – Forcing users to change passwords
  - ✓NIST's 2003 password guidelines recommend regular (possibly monthly) password updates.
  - ✓Later security usability researchers found out that over 40% of passwords could be guessed from previous ones; forced change did not do much help users not choose weak passwords. (They tend to choose password, e.g., alice04 (for April), alice05 (for May) etc.)
  - ✓**NIST now recommends long passphrases that are changed on compromise.**