



Passwords | Time-memory trade-offs

INGI2347: COMPUTER SYSTEM SECURITY (Spring 2015)

Marco Canini | Guest lecturer: Xavier Carpent

UCL
Université
catholique
de Louvain



Plan for today

Lecture 12

■ Passwords

- Vulnerabilities
- Online Attacks
- Offline Attacks
- Weak Passwords
- Unix/Windows Cases
- Strong Passwords and Good Practices

■ Time-memory trade-offs

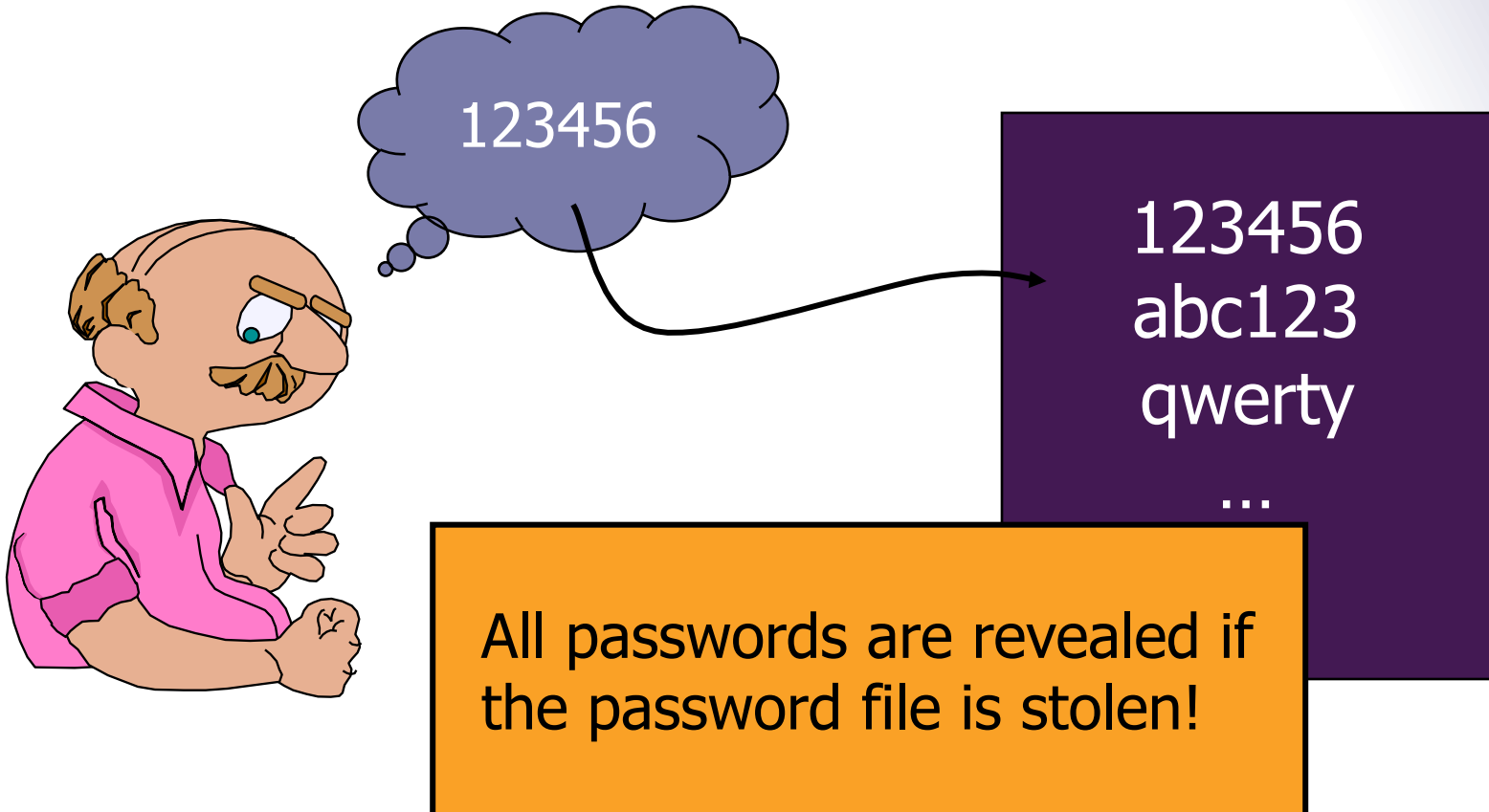




Naïve Idea

User

Password file





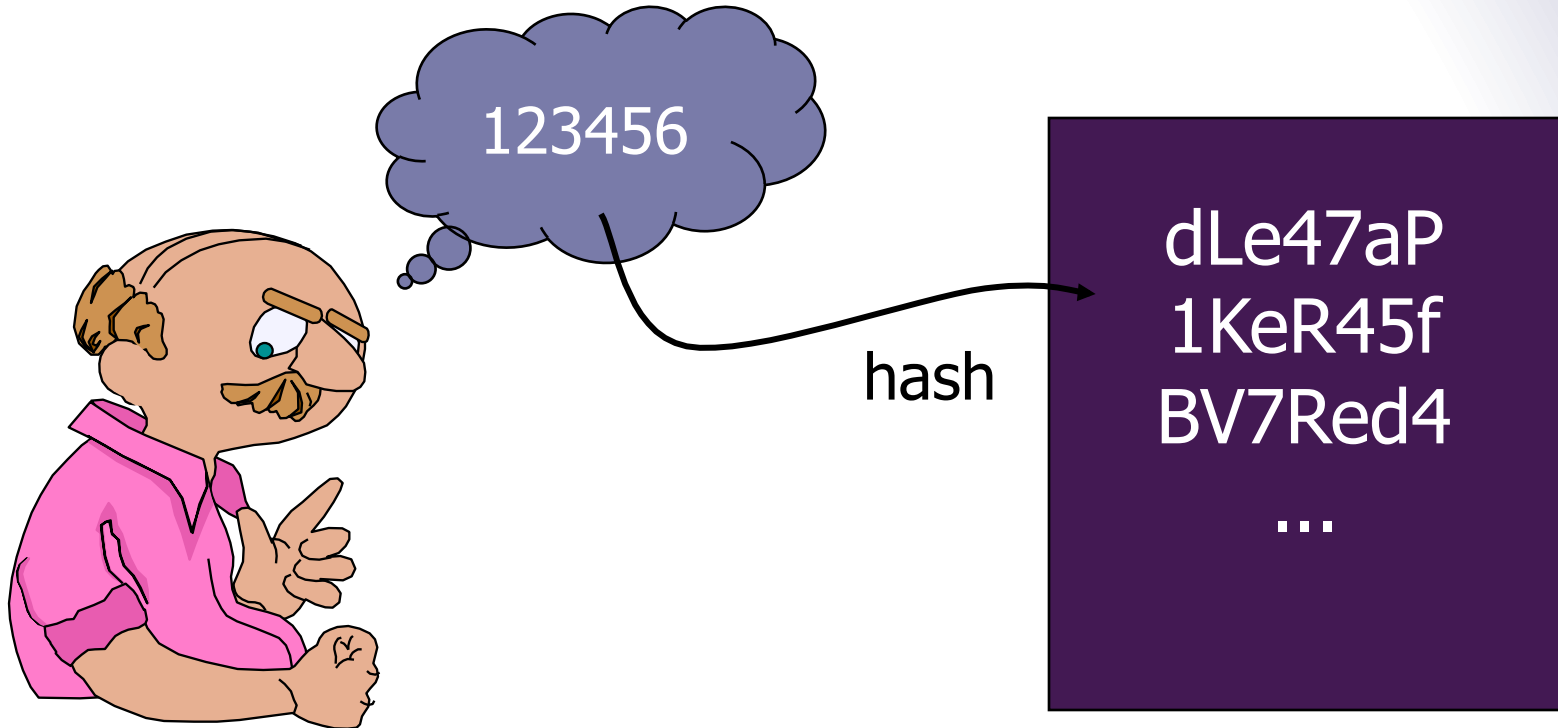
Password Storage

- Passwords must never be stored as plaintext!
- Instead of passwords, store a hash
- The hash must be irreversible
- When logging in, the hashed password is compared with the stored hash

Implemented Idea

User

Hash file





Vulnerabilities

+ Some Vulnerabilities

- Written down passwords
- Shoulder surfing
- Social engineering
- Key logger, Rootkit
- Eavesdropping the network
- Multi-website passwords
- Audit trails
- Guessing the password (low entropy)



Written Down Passwords

According to a 2002 security survey:

- Probability of finding written passwords near a computer subjected to periodic password changes varied from 16% to 39%
- Probability varied from 4% to 9% when the administrator did not enforce periodic password changes



Shoulder Surfing

- Password keystroke **observed**
 - E.g. camera above an ATM
- Graduate students at the University of Maryland Baltimore County shown that:
 - Non-dictionary passwords are **more vulnerable** to shoulder surfing than passwords belonging to a dictionary
- Some keys are more easily **observable**



Social Engineering

- Abuse the users

Survey at AArhus University:

- 336 students were asked by mail to send back their passwords to validate the password database
- 138 revealed their passwords
- A few changed their passwords, but no one reported to the system administrator

Key Logger, Rootkit

- Software or Hardware
- Program that runs in the background, recording all the keystrokes.
- Device between the keyboard and the computer
 - It has a microcontroller and a non-volatile memory
 - Microcontroller interprets the keystrokes as they are typed and stores them in the memory
- Software example: ActualSpy

Key Logger, Rootkit

- **Solution:** On-screen keyboard, password typed in different order using the mouse

AUTH - Mozilla Firefox

File Edit View History Bookmarks Tools Help

https://www.normand-g2-enligne.credit-agricole.fr/

Customize Links Free Hotmail Windows Marketplace Windows Media Windows

NORMANDIE

mercredi 9 avril 2008

IDENTIFICATION [?]

Le Crédit Agricole accorde une grande attention à la sécurité de ses sites Internet. En tant qu'utilisateur, vous avez également un rôle important et actif à jouer pour assurer la sécurité des informations qui vous concernent. Consultez ci-dessous nos recommandations :
» Nos recommandations

VOS CODES D'ACCES

Saisissez votre N° DE COMPTE à l'aide de votre clavier :

Cliquez dans la grille pour composer votre CODE PERSONNEL :
(6 chiffres)

» Voir la démonstration

Je reconnais avoir pris connaissance et accepté les conditions juridiques de la convention d'utilisation du service.
» Conditions juridiques

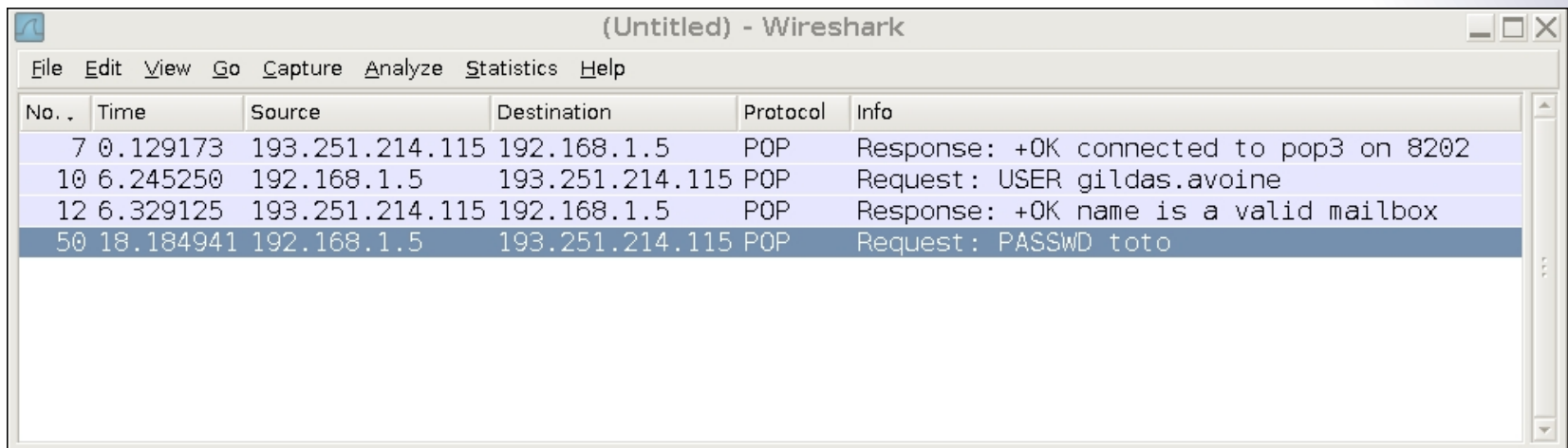
©2004-2007 Crédit Agricole, tous droits réservés

Find: avoine Next Previous Highlight all Match case

Done www.normand-g2-enligne.credit-agricole.fr

Eavesdropping the network

- Passwords sent in the clear through the network: POP, FTP



The image shows a screenshot of the Wireshark network protocol analyzer interface. The title bar reads "(Untitled) - Wireshark". The menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, and Help. The main display area shows a list of captured packets with the following columns: No., Time, Source, Destination, Protocol, and Info. Four packets are visible, all of POP protocol type, showing a successful login sequence where the password "toto" is transmitted in plaintext.

No.	Time	Source	Destination	Protocol	Info
7	0.129173	193.251.214.115	192.168.1.5	POP	Response: +OK connected to pop3 on 8202
10	6.245250	192.168.1.5	193.251.214.115	POP	Request: USER gildas.avoine
12	6.329125	193.251.214.115	192.168.1.5	POP	Response: +OK name is a valid mailbox
50	18.184941	192.168.1.5	193.251.214.115	POP	Request: PASSWD toto

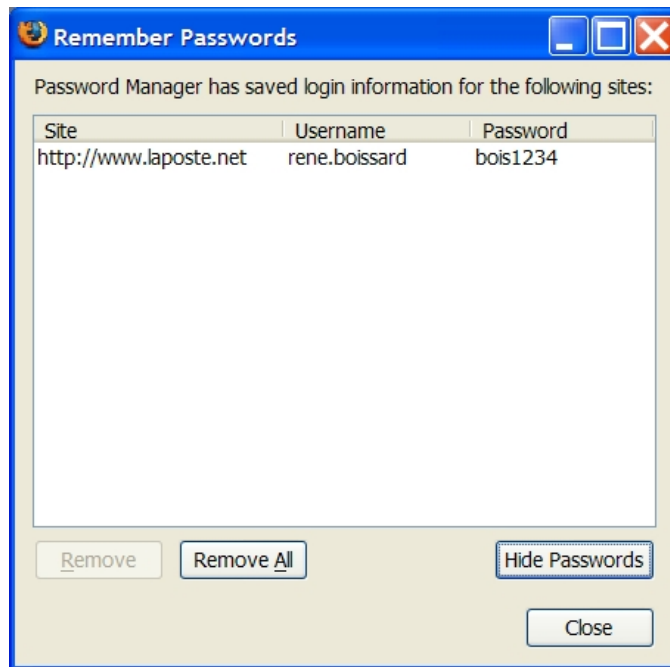
A POP session sniffed with Wireshark

Multi-Website Passwords

- Passwords should never be used for different purposes
 - Never use the same password for both Windows and Unix
 - Never use a password received by email for secure applications
- A common practice is to use different security level passwords
 - Good different passwords for Windows accounts, Unix accounts, main mailbox
 - A few weaker passwords (easier to remember) for less secure applications, like online registration with pseudo

Audit Trails

- Audit Trails can reveal the user name of the users
 - Password managers (be careful on public computers)
 - People enter passwords in the field of user name
 - Passwords in emails





Guessing some Password(s)

- Targeted attack on **one** account
- Attempt to penetrate **any** account **on a** system

Guessing a (the) Password(s)

■ Online Attack

- The system is used as an oracle (black box)
- Slow

■ Offline Attack

- The attacker steals the hash file
- The attacker recovers the passwords offline
- The algorithm must be known

■ Target

- A given account
- Any account on the system

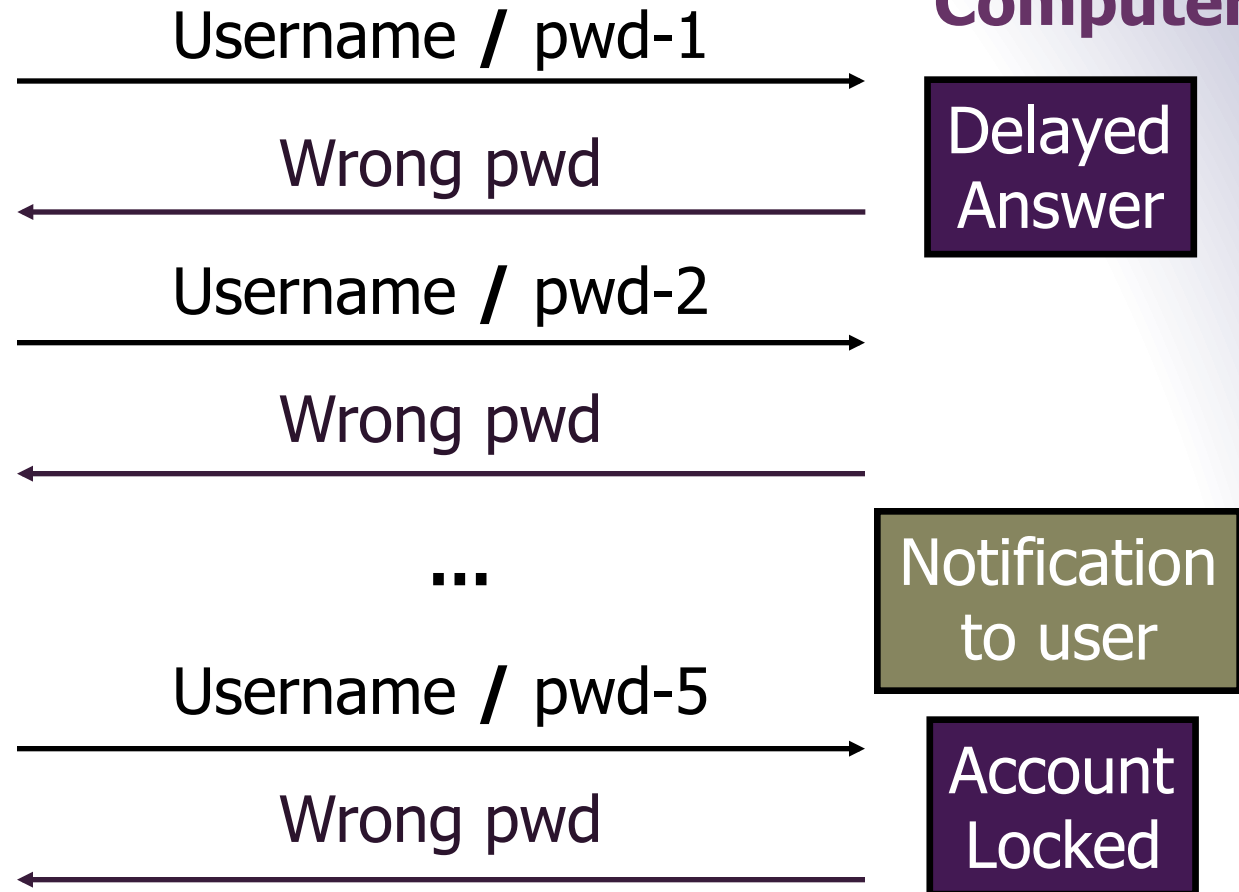


Online Attacks

Countermeasures

User

Computer





Locking Account

Denial of service attacks:

- To lock a user, try to login into his account with random passwords

Customer service costs:

- Users whose accounts are locked call a customer service center

Computing Cost for the User

- Each login attempt must be accompanied by $h(\text{username}, \text{pwd}, r)$ such that 20 least significant bits are 0
- Negligible overhead for a single request
- Attacks are slowed
- Implementation Issues:
 - Clients must use a special software
 - Legitimate user with a slow machine

Captcha

- Legitimate logins are done by **humans** while attacks are done by **computers**
- **Captcha**: Completely **A**utomated **P**ublic Turing Test to tell **C**omputers and **H**umans **A**part
- Login attempts must be accompanied by a **computation** that is **easy** for humans and **hard** for programs





Offline Attacks

Offline Cracking

- Hash algorithm must be known
- Attacker must obtain a copy of passwords' hashes
- Since she cannot inverse hashes, she must guess the passwords (dictionary) or perform an exhaustive search
- She generates the hashes of those words
- She finally compares the generated hashes with the stolen hashes until finding a match

Dictionary Attacks

- Many people use **dictionary words** as passwords
 - Average dictionary contains only 150,000 to 200,000 words
 - People's names, common pet names, and ordinary words
- Hence files containing hashed passwords are susceptible to **pre-compiled dictionary attack**
 - A file of hashes of all possible dictionary words is generated
- A PC can generate **200,000 to 10,000,000** password hashes per second depending on the type of hash

Heuristic Attack

- Combine dictionary and brute force
- Some rules are applied to the dictionary words according to the most used practices
 - Convert to lowercase, uppercase
 - Capitalize
 - Reverse: "Fred" -> "derF"
 - Duplicate: "Fred" -> "FredFred"
 - Reflect: "Fred" -> "FredderF"
 - Rotate the word left: "jsmith" -> "smithj"
 - Rotate the word right: "smithj" -> "jsmith"
 - Append or prefix character X to the word
 - Prefix the word with character X



Offline Attack Procedure

Progressive cracking:

- Trivial and short passwords
- Dictionary + Heuristics
- Brute force

Cracking Tools:

- Unix/Windows cracking: John the ripper, L0phtCrack
- Windows password cracking: Cain, Ophcrack



Weak Passwords



Weak Passwords

- Based on common dictionary words
- Based on common names
- Based on user/account identifier
- Short (under 7 characters)
- Based on keyboard patterns (e.g., “qwerty”)
- Composed of single symbol type (e.g., characters)
- ...



Weak Passwords: Length

Length	Percent
1-4	0.82%
5	1.1%
6	15%
7	23%
8	25%
9	17%
10	13%
11	2.7%
12	0.93%
13-32	0.93%

Source: www.schneier.com



Weak Passwords: Content

numbers only	1.3%
letters only	9.6%
alphanumeric	81%
non-alphanumeric	8.3%

Source: www.schneier.com

Weak Passwords

- Top-used passwords are (in order):

password1, abc123, myspace1, password, blink182, qwerty1, fuckyou, 123abc, baseball1, football1, 123456, soccer, monkey1, liverpool1, princess1, jordan23, slipknot1, superman1, iloveyou1, monkey.

Source: www.schneier.com

- “We used to quip that ‘password’ is the most common password. Now it’s ‘password1’. Who said users haven't learned anything about security?” (Schneier, 2006)
- Passwords are much better today than 15 years ago



Unix/Windows Cases



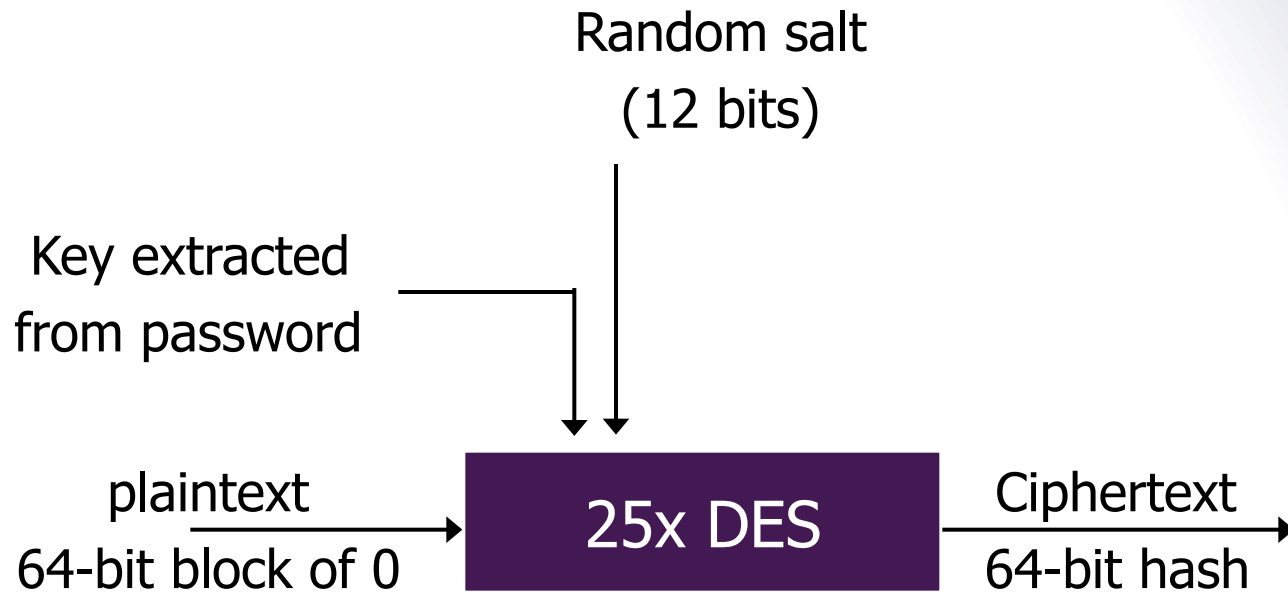
Unix Passwords

The hash function can be based on:

- DES
- MD5 (Linux, BSD, Sun)
- Blowfish (OpenBSD)
- SHA256
- SHA512

+ Unix Passwords (DES)

35





Unix Passwords (MD5)



Storage Under Unix

■ Old method:

- Name and hashes of passwords in the file `/etc/passwd` with free read access

■ Safer method:

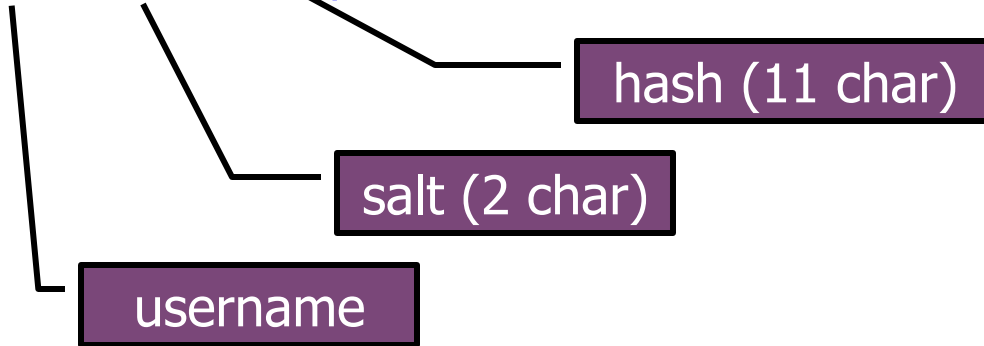
- The hashes are found in a separate file, `/etc/shadow` that can be read only by the administrator
- Why is it safer since the function is one-way?

■ Two ways to gain access to the password file:

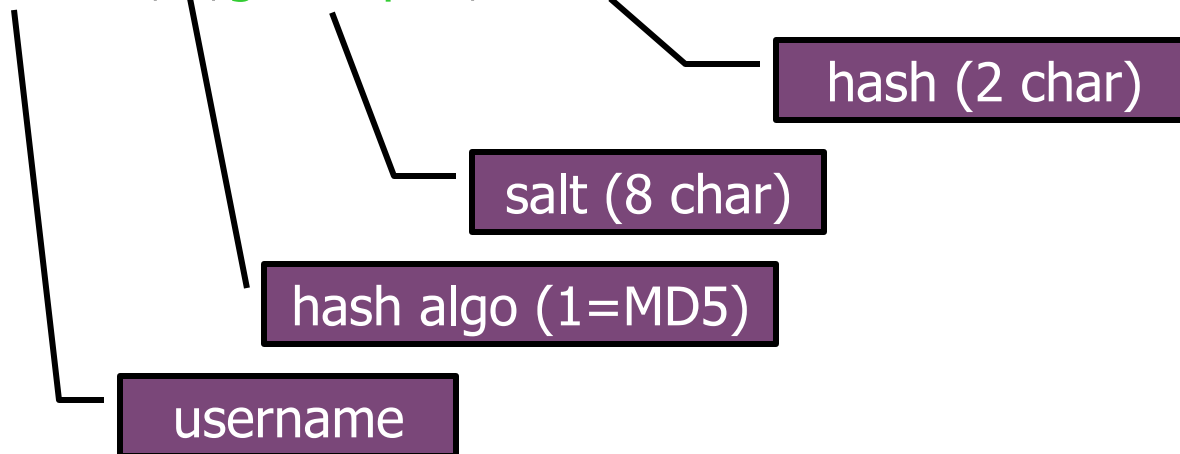
- Reboot the machine with a USB key or a CD
- Obtain administrator privileges using an exploit

/etc/shadow (DES, MD5)

Smith:3Yr83xxCi/Ki2:12801:0:99999:7:-1::



Smith:\$1\$gDT4Spf5\$mr76vshidvcT1busoKrre1:11001:0:99999





Practice Yourself

DES:

- `openssl passwd -crypt -salt pH <PASSWORD>`

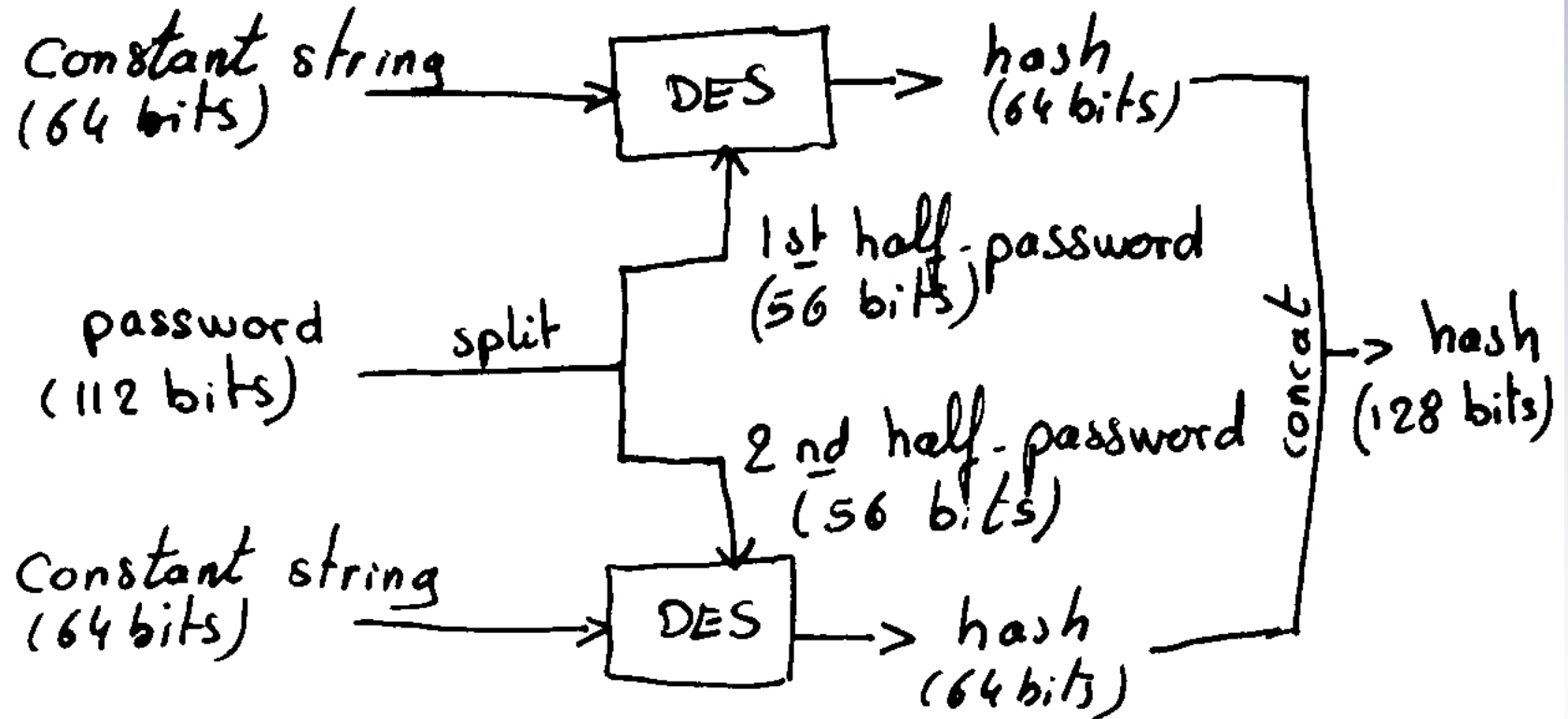
MD5:

- `openssl passwd -1 -salt gDT4Spf5 <PASSWORD>`

Win 9x Passwords (LM Hash)

- Win98/ME uses the Lan Manager Hash (LM hash)
- The password is cut in two blocks of 7 characters after completion to 14 characters with empty char
- Lowercase letters are converted to uppercase
- A separate hash is generated for each 7-char block
- The 7 bytes block are used as DES keys to encrypt an 8-byte constant string:
 - 0x4B, 0x47, 0x53, 0x21, 0x40, 0x23, 0x24, 0x25
- The LM hash does not use any salt
- <http://lasecwww.epfl.ch/~oechslin/projects/ophcrack>

Win 9x Passwords (LM Hash)





Win NT/2000/XP/Vista/Seven (NT LM Hash)

- Win NT/2000/XP/Vista/Seven uses the NT Lan Manager Hash (aka NT hash)
- The password is **no** longer **cut** in two blocks
- Passwords can be longer than 14 characters (but compatibility issues arise beyond 14 characters)
- Lowercase letters **are not converted** to uppercase
- The hash function is **MD4**
- The NT hash still **does not use any salt**

Storage

- Under W2k, XP, 2003, NTLM and LM hash of all users are stored in the **Security Account Manager** file or in the **Active Directory** (ntds.dit)
- The file is encrypted, but by default the **key** can be **extracted from the machine**
- If the machine is running we need **administrator** privileges plus a special exploit (**pwdump**) to extract the hashes
- If we can **boot another OS**, we can steal and decrypt the hashes




Cracking Times – Benchmarks John (2011)

- Traditional DES: 1134K c/s
- FreeBSD MD5: 4400 c/s
- OpenBSD Blowfish: 269 c/s
- LM DES: 6547K c/s
- NT MD4: 8260K c/s



LM Hash

- All (LM Hash) alphanum passwords cracked within a few seconds (success 99.9%)
- (Alphanum + 15 special char) LM Hash passwords cracked in a few minutes (success about 96%)
- Storage: CD or DVD (fit the RAM)
- See <http://ophcrack.sourceforge.net/>



+ Strong passwords and good practices

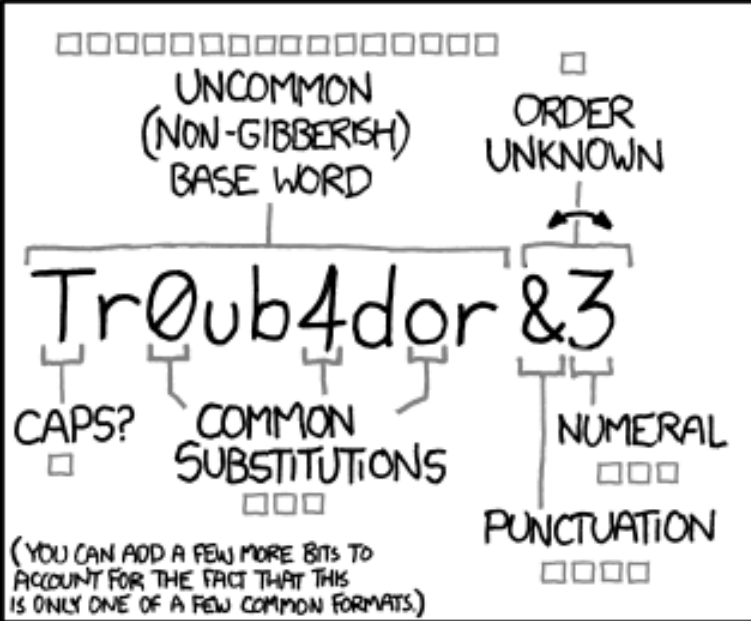


Strong Passwords

- Contain at least *one of each* of the following
 - Digit (0...9)
 - Letter (a...Z)
 - Punctuation symbol (e.g., !)
 - Control character (e.g., ^s, Ctrl-s)
 - Special character in the first 7 characters
- Based on a verse (e.g., passphrase)
- Easily remembered but difficult for others to guess

+ Some good practices

- Never recycle passwords
- Never record a password anywhere
 - Exceptions include encrypted password “vaults”
- Use a different password for each system/context
- Change password regularly (?)
- Change your password immediately if you suspect it has been “stolen”, or after using a public computer
- Passwords should be protected in a manner that is consistent with the damage that could be caused by their compromise



~28 BITS OF ENTROPY

$2^{28} = 3 \text{ DAYS AT } 1000 \text{ GUESSES/SEC}$

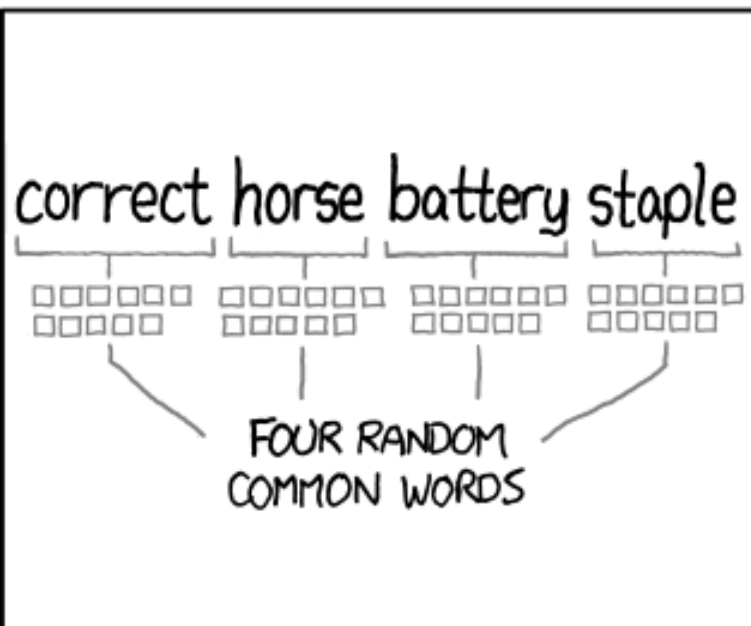
(PLAUSIBLE ATTACK ON A WEAK REMOTE WEB SERVICE. YES, CRACKING A STOLEN HASH IS FASTER, BUT IT'S NOT WHAT THE AVERAGE USER SHOULD WORRY ABOUT.)

DIFFICULTY TO GUESS: **EASY**

WAS IT TROMBONE? NO, TROUBADOR. AND ONE OF THE 0s WAS A ZERO?

AND THERE WAS SOME SYMBOL...

DIFFICULTY TO REMEMBER: **HARD**



~44 BITS OF ENTROPY

$2^{44} = 550 \text{ YEARS AT } 1000 \text{ GUESSES/SEC}$

DIFFICULTY TO GUESS: **HARD**

THAT'S A BATTERY STAPLE.

CORRECT!

DIFFICULTY TO REMEMBER: YOU'VE ALREADY MEMORIZED IT

THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.