Authentication is the process of verifying an identity claimed by or for a system entity.

Authorization is the function of specifying access rights to resources related to information security and computer security in general and to access control in particular.
Means/Factors of Authentication

- Something the individual knows
- Something the individual possesses
- Something the individual is/does
Something the User Knows

**Password**
- As56kf#dfjd8%d
- John123
- JustinBieber14
- Y3llow5ubm4rine

**PIN**
- 123456
- 654321
- 1248
- 338

**Answers (to questions)**
- What is the name of your dog?
- What is your favorite color?
- What... is the air-speed velocity of an unladen swallow?
Acquaintance with whom participants reported being unwilling to share their webmail passwords were able to guess 17% of their answers.

Participants forgot 20% of their own answers within six months.

... 13% of answers could be guessed within five attempts by guessing the most popular answers of other participants ...

It's no secret: Measuring the security and reliability of authentication via 'secret' questions

United Mileage Plus

Yesterday, Yan Zhu (@bcrypt) pointed out on Twitter that United Airlines Mileage Plus program has started collecting answers to security questions. They have a new twist: you must select one of a menu of answers.

United wants the answers to five questions, chosen from a list:
Something the User Possesses

- Smart card
- Ring
- Key
- USB drive
- Mobile phones
- SecureID device
Something the Individual...

..Is

..does
Something the Individual...

..Is

..does

NOT just face recognition
Something the Individual...

..Is

..does

Spring 2018
Stevens Institute of Technology
Something the Individual...

..Is

How about CAPTCHA?

"Completely Automated Public Turing test to tell Computers and Humans Apart"
Password Authentication
Passwords

Widely used

Process

- User provides name/login and password
- System compares password with the one stored for that specified login

The user ID:

- Determines that the user is authorized to access the system
- Determines the user’s privileges
- Is used in discretionary access control
Passwords Naïve Implementation

Non-confidential channel

username: bob
password: p4ssw0rd
Passwords Naïve Implementation

Non-confidential channel

username: bob
password: p4ssw0rd

Eavesdropper
Passwords Naïve Implementation

Non-confidential channel

Confidential channel

username: bob
password: p4ssw0rd
Passwords Naïve Implementation

Non-confidential channel

Confidential channel

username: bob
password: p4ssw0rd

Password DB leak
Passwords Naïve Implementation

Non-confidential channel

Confidential channel

username: bob
password: p4ssw0rd

Insider

Spring 2018

Stevens Institute of Technology
### Password Leaks Happen All the Time

<table>
<thead>
<tr>
<th>Year</th>
<th>Company</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>2009</td>
<td>RockYou Gaming</td>
<td>32.0 million</td>
</tr>
<tr>
<td>2010</td>
<td>Gawker Media</td>
<td>1.5 million</td>
</tr>
<tr>
<td></td>
<td><em>Domino attack prompted resets in other sites</em></td>
<td></td>
</tr>
<tr>
<td>2011</td>
<td>Sony</td>
<td>1.0 million</td>
</tr>
<tr>
<td></td>
<td><em>Before being detected and shut down</em></td>
<td></td>
</tr>
<tr>
<td>2012</td>
<td>LinkedIn</td>
<td>6.5 million</td>
</tr>
<tr>
<td>2013</td>
<td>Twitter</td>
<td>250,000</td>
</tr>
<tr>
<td>2013</td>
<td>Adobe</td>
<td>150.0 million</td>
</tr>
<tr>
<td>2015</td>
<td>Ashley Madison</td>
<td>15.26 million</td>
</tr>
</tbody>
</table>
Hashed Passwords

Password File

User ID | Hash code
---|---

slow hash function

Load

Password

Spring 2018
Stevens Institute of Technology
Hash Function Requirements

Can be applied to a block of data of any size

Produces a fixed-length output

H(x) is relatively easy to compute for any given x

Computationally infeasible to find x such that H(x) = h

Computationally infeasible to find y ≠ x such that H(y) = H(x)

Computationally infeasible to find any pair (x,y) such that H(x) = H(y)
Security of Hash Functions

There are two approaches to attacking a secure hash function:

- **Cryptanalysis**: Exploit logical weaknesses in the algorithm
- **Brute-force attack**: Strength of hash function depends solely on the length of the hash code produced by the algorithm

MD5 and SHA-1 have been broken through cryptanalysis.
SHA-2 or later is suggested.
Adding Salt

Password File

<table>
<thead>
<tr>
<th>User ID</th>
<th>Salt</th>
<th>Hash code</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

slow hash function

Salt

Password

Load
Hashed Passwords Today

Non-confidential channel

Confidential channel

username: bob
password: p4ssw0rd

Password DB leak

H(p4ssw0rd)

Insider

Spring 2018
Stevens Institute of Technology
Password Cracking

Dictionary attacks

Brute-force

Combination of the above

John the Ripper – first open-source password cracker developed in 1996
Dictionary Attacks

Develop a large dictionary of possible passwords and try each against the password file

Each password must be hashed using each salt value and then compared to stored hash values

Good dictionaries and heuristics for combining words give attackers an advantage.

Publicly available databases of cracked passwords also help.
Rainbow Table Attacks

Pre-compute tables of hash values for all salts

A mammoth table of hash values

Can be countered by using a sufficiently large salt value and a sufficiently large hash length

Researchers have shown that using 1.4 GB of data, they could crack 99.9% of all alphanumeric Windows password hashes in 13.8 seconds.
Percentage of Passwords Guessed

Using DB of leaked password files, including the RockYou file.
An online password cracking service for penetration network auditors who need to check the security of wireless networks, crack password hashes, and decrypt encryptions.

http://hashcat.net/oclhashcat/

Download latest version

<table>
<thead>
<tr>
<th>Name</th>
<th>Version</th>
<th>md5sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>oclHashcat for AMD</td>
<td>v1.30</td>
<td>4d6e7baddb35d34f77455b5350a</td>
</tr>
<tr>
<td>oclHashcat for Nvidia</td>
<td>v1.30</td>
<td>1e7de6927c674c560af20693373aa</td>
</tr>
</tbody>
</table>

GPU Driver requirements:
- NV users require ForceWare 331.67 or later
- AMD users require Catalyst 14.6b or later

Features
- Worlds fastest password cracker
- Worlds first and only GPGPU based rule engine
- Free
- Multi-GPU (up to 128 gpus)
- Multi-Hash (up to 100 million hashes)
- Multi-OS (Linux & Windows native binaries)
- Multi-Platform (OpenCL & CUDA support)
- Multi-Algo (see below)
- Low resource utilization, you can still watch movies or play games while cracking
- Focuses highly iterated modern hashes
- Focuses dictionary based attacks
- Supports distributed cracking
- Supports pause / resume while cracking
- Supports sessions
- Supports restore
- Supports reading words from file
- Supports reading words from stdin
- Supports hex-salt
- Supports hax-charsalt
- Built-in benchmarking system
- Integrated thermal watchdog
- 100+ Algorithms implemented with performance in mind
- ... and much more

Hashcat Screenshot

Device #1: Hawaii, 3072MB, 1000MHz, 44MCU
Hashes: 1 hashes, 1 unique digests, 1 unique salts
Bitmaps: 8 bits, 256 entries, 0x0000000f mask, 1024 bytes
Applicable Optimizers:
- Zero-Byte
Welcome to Radeon City, population: 6. It's one of five servers that make up a high-performance password-cracking cluster.

A password-cracking expert has unveiled a computer cluster that can cycle through as many as 350 billion guesses per second. It's an almost unprecedented speed that can try every possible Windows password in the typical enterprise in less than six hours.
HoneyPasswords

Leak and crack passwords

1/N chances to log in “legitimately”

N decoy passwords

... JohnP:mypassword
JohnP:mypassword1
JohnP:password
JohnP:thepassword
...

Site A

... jp:mypassword
...

Site B

Spring 2018
Stevens Institute of Technology
Other Threats

- Password reuse
- Social Engineering
- Phishing
Phishing

Connect with your friends faster, wherever you are.

The Facebook application is available in more than 2,500 phones.

- Faster navigation
- Compatible with the camera and your phone contacts
- Without regular updates: download only

Discover Facebook Mobile
Synergistic Authentication (Sauth)

Most users login in multiple web services

...and they stay logged in

Exploit this to protect ourselves from attackers that have obtained our password
Services Can Vouch for the User

User Agent

1a) authentication as $A \rightarrow S$

1b) redirection to $V$ for vouching

3a) vouching response verification

3b) access to the $A \rightarrow S$ account

Service $S$

(e.g., Twitter)

Service $V$

(e.g., Google)

(2b) redirection to $S$ with vouching response

(2a) authentication as $A \rightarrow V$
Password Alternatives
Graphical Passwords
Social Authentication

Why is this hard?

“Social Authentication: Harder than it Looks”
https://www.cl.cam.ac.uk/~rja14/Papers/socialauthentication.pdf
Authentication with Insecure Communication

n\textsuperscript{th} password $\rightarrow$ $H^n = n \times \ldots \times H(H(“p4ssw0rd”))$

Server asks for n password

Calculate and send $H^n$

password: p4ssw0rd

n = 1000

password: $H^{1000}$

Compare hashed passwords

n = 999

...
Lamport’s Hash

When $n == 0$ password needs to be reset

No mutual authentication

Still vulnerable to MiTM

Why?
Authentication with Insecure Communication

Leslie Lamport, Password Authentication with Insecure Communication, 1981

\[ n^{\text{th}} \text{ password } \rightarrow H^n = n \text{ times } \ldots H(H("p4ssw0rd")) \]

MiTM asks for n-M password

Server asks for n password

Calculate and send \( H^{n-M} \)

Calculate and send
Tokens
Memory Cards

Can store but do not process data

The most common is the magnetic stripe card

Can include an internal electronic memory

Can be used alone for physical access
  - Hotel room
  - ATM

Provides significantly greater security when combined with a password or PIN

Drawbacks of memory cards include:
  - Requires a special reader
  - Can be stolen
  - User needs to carry them
Token-based Authentication

Channel requires contact or close proximity

Spring 2018
Stevens Institute of Technology
**Smart Tokens**

**Physical characteristics:**
- Include an embedded microprocessor
- A smart token that looks like a bank card
- Can look like calculators, keys, small portable objects

**Interface:**
- Manual interfaces include a keypad and display for interaction
- Electronic interfaces communicate with a compatible reader/writer

**Authentication protocol:**
- Classified into three categories:
  - Static
  - Dynamic password generator
  - Challenge-response
Static Protocol

Channel requires contact or close proximity

E(TOKEN) -> TOKEN

Key

Spring 2018
Stevens Institute of Technology
Dynamic Protocol

Time-based One Time Password Generation

Valid for a limited amount of time

SECRET

OTP

SECRET

Spring 2018
Stevens Institute of Technology
Simple Mutual Authentication (Challenge-Response)

secret = H(password)

Client sends $cr + cc$

Server sends $sc$

Server sends $sr$

Generate unique random value $cc$ (nonce)

Generate $cr$ and check received value

Generate $sr = H(sc + cc + secret)$

Generate unique random value $cc$

and calculate $cr = H(cc + sc + secret)$

Generate $sr$ and check received value
Simple Mutual Authentication (Challenge-Response)

secret = H(password)

Generate unique random value \( cc \)
and calculate
\[ cr = H(cc + sc + secret) \]

Generate \( sr \) and check received value

Server sends \( sc \)

Client sends \( cr + cc \)

Server sends \( sr \)

Generate unique random value \( sc \) (nonce)

Generate \( cr \) and check received value

Generate \( sr = H(sc + cc + secret) \)
Challenge-Response Protocol

Using public-key cryptography

- Secret key $PK^+$
- Public key $PK^-$

Client sends $cr + \text{SIG}(PK^+, cr)$

Server sends $sc + \text{SIG}(PK^-, sc)$

Generate unique random value $sc$ (nonce)

Verify signature

Generate unique random value $cc$

Verify signature
Security Issues with Cards

Information may be unencrypted on the card

They can be reverse engineered
Cracking the Mifare Chip

https://www.youtube.com/watch?v=NW3RGbQTLhE
Biometric Authentication

Attempts to authenticate an individual based on unique physical characteristics

Based on pattern recognition

Is technically complex and expensive when compared to passwords and tokens

Physical characteristics used include:

- Facial characteristics
- Fingerprints
- Hand geometry
- Retinal pattern
- Iris
- Signature
- Voice
Cost vs Accuracy for Biometrics

Cost
- Hand
- Signature
- Face
- Voice

Accuracy
- Iris
- Retina
- Finger

Spring 2018
Stevens Institute of Technology
Enrollment creates an association between a user and the user's biometric characteristics. Depending on the application, user authentication either involves verifying that a claimed user is the actual user or identifying an unknown user.
Probabilistic Identification

Matching score ($s$)

- Probability density function
- Imposter profile
- False nonmatch possible
- Average matching value of imposter
- Decision threshold ($t$)
- Profile of genuine user
- False match possible
- Average matching value of genuine user

In this depiction, the comparison between presented feature and a reference feature is reduced to a single numeric value. If the input value ($s$) is greater than a preassigned threshold ($t$), a match is declared.
Figure 3.10   Idealized Biometric Measurement

Operating Characteristic Curves (log-log scale)

- Increase threshold: increased security, decreased convenience
- Decrease threshold: decreased security, increased convenience

Equal-error rate (ERR) is an important metric. Lower is better.
Figure 3.11  Actual Biometric Measurement Operating Characteristic Curves, reported in [MANS01]. To clarify differences among systems, a log-log scale is used.
Location as a 4th Factor
Location-Based Verification Using Smartphones

Are you at location? Yes (or No)

Transaction OK

Processor
Location-Based Verification

Advantages
79% of people aged 18–44 have their smartphones with them 22 hours a day

Disadvantages
It’s not 100%
- May forget phone
- Phone can run out battery
- May leave phone behind during certain activities (e.g., running in the park)
Access Control
High-level Overview

User → Reference monitor

Access request → Resources

Policy
Access Control Approaches

Discretionary Access Control (DAC)
- Resources are usually associated with an owner
- Discretionary because the owner can delegate access

Mandatory Access Control (MAC)
- Operating system or reference monitor strictly manages access
- Access can not be delegated
## DAC Example: UNIX Permissions

<table>
<thead>
<tr>
<th>Entities</th>
<th>User owner</th>
<th>Group</th>
<th>Others</th>
</tr>
</thead>
</table>

### Resource

- **File Object**

### Access type

- **RWX**
- **RWX**
- **RWX**
### MAC Example: Access control list (ACL)

<table>
<thead>
<tr>
<th>Entity</th>
<th>Access type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Resource

File Object
Role-based Access Control (RBAC)

Policies apply on roles
- Roles are similar to groups

Usually less roles than users → easier management

Easy to handle users switching roles

DAC or MAC afterwards
Role Hierarchy

More rights:
- Administrator
- PowerUser
- User
- Guest

Less rights:
Mix and Match

In practice multiple approaches are usually combined to control different type of requests and resources
Additional Reading

The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes

Social Authentication: Harder than it Looks

Honeywords: Making Password-Cracking Detectable

SAuth: Protecting User Accounts from Password Database Leaks

Smartphones as Practical and Secure Location Verification Tokens for Payments

Dos and Don’ts of Client Authentication on the Web

Kerberos: The Network Authentication Protocol