Lecture #2: Defence Strategies and Techniques: Part I

Dr.Ramchandra Mangrulkar

August 5, 2020

Outline...

- Access Control
- Data Protection
- Prevention and Detection
- Response, Recovery and Forensics

Access Control

- First Strategy
- Security technique that regulates who or what can view
- Selective restriction of access to a place or other resource
- Fundamental concept in security that minimizes risk to the business or organization.
- Two Types : Physical and Logical.
- Physical access control limits access to campuses, buildings, rooms and physical IT assets.
- Logical access control limits connections to computer networks, system files and data.

Access Control

- First Strategy
- Security technique that regulates who or what can view
- Selective restriction of access to a place or other resource
- Fundamental concept in security that minimizes risk to the business or organization.
- Two Types : Physical and Logical.
- Physical access control limits access to campuses, buildings, rooms and physical IT assets.
- Logical access control limits connections to computer networks, system files and data.

Access Control

- First Strategy
- Security technique that regulates who or what can view
- Selective restriction of access to a place or other resource
- Fundamental concept in security that minimizes risk to the business or organization.
- Two Types : Physical and Logical.
- Physical access control limits access to campuses, buildings, rooms and physical IT assets.
- Logical access control limits connections to computer networks, system files and data.

Access Control

- First Strategy
- Security technique that regulates who or what can view
- Selective restriction of access to a place or other resource
- Fundamental concept in security that minimizes risk to the business or organization.
- Two Types : Physical and Logical.
- Physical access control limits access to campuses, buildings, rooms and physical IT assets.
- Logical access control limits connections to computer networks, system files and data.

Access Control

- First Strategy
- Security technique that regulates who or what can view
- Selective restriction of access to a place or other resource
- Fundamental concept in security that minimizes risk to the business or organization.
- Two Types : Physical and Logical.
- Physical access control limits access to campuses, buildings, rooms and physical IT assets.
- Logical access control limits connections to computer networks, system files and data.

Access Control

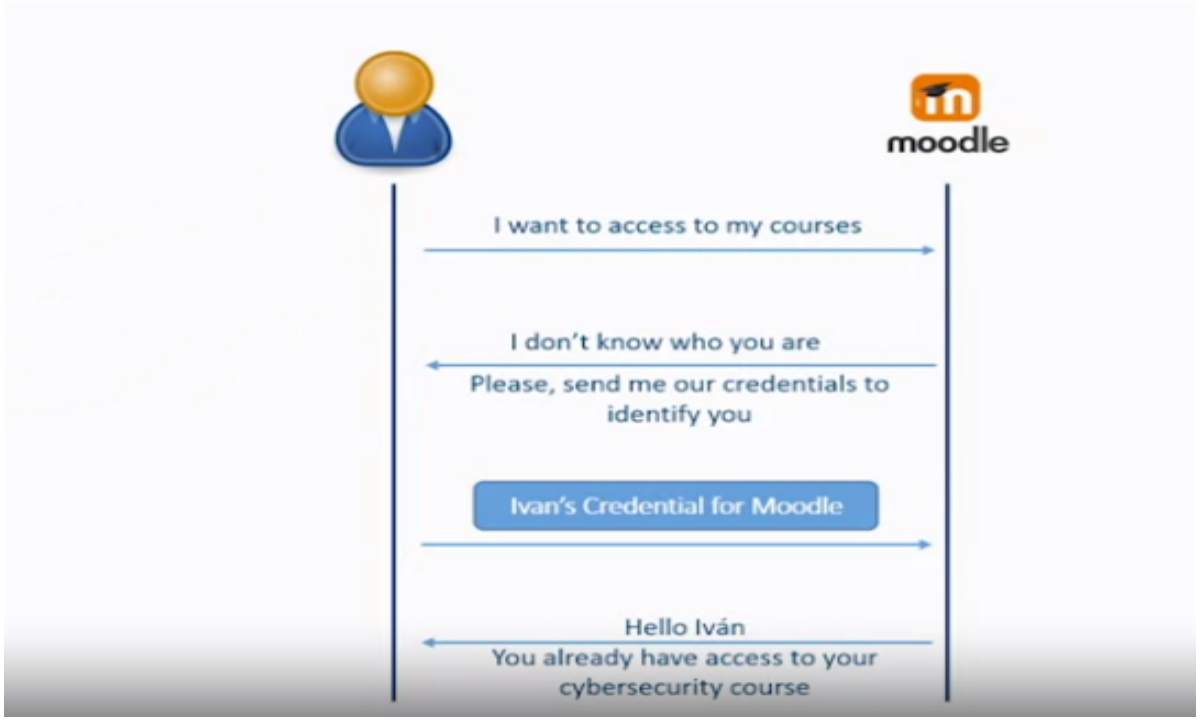
- First Strategy
- Security technique that regulates who or what can view
- Selective restriction of access to a place or other resource
- Fundamental concept in security that minimizes risk to the business or organization.
- Two Types : Physical and Logical.
- Physical access control limits access to campuses, buildings, rooms and physical IT assets.
- Logical access control limits connections to computer networks, system files and data.

Access Control

- First Strategy
- Security technique that regulates who or what can view
- Selective restriction of access to a place or other resource
- Fundamental concept in security that minimizes risk to the business or organization.
- Two Types : Physical and Logical.
- Physical access control limits access to campuses, buildings, rooms and physical IT assets.
- Logical access control limits connections to computer networks, system files and data.

Access Control(Continue...)

- Authentication and Authorization[1]

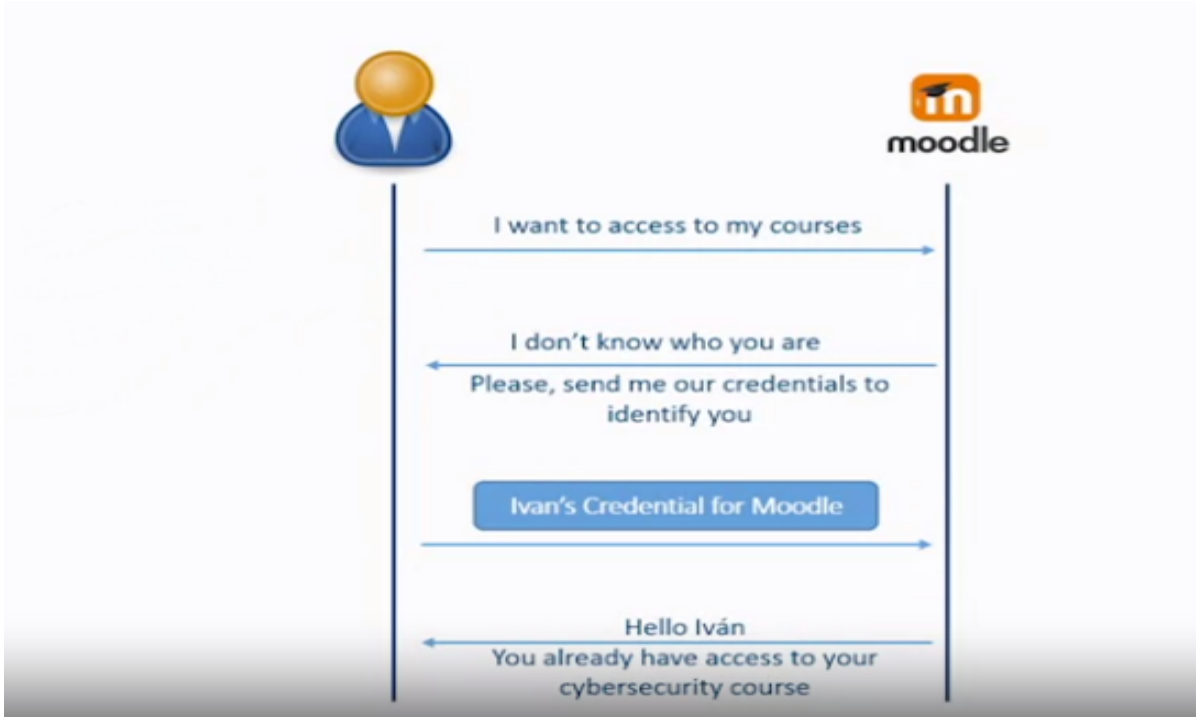


- Authentication Methods (Three-factor authentication) :
- Something you know (Something I know)
- Something you have (Something I Have)
- Something you are (Something I am)

[1] https://www.coursera.org/learn/cybersecurity/lecture/sQpu1/introduction-to-identity-credentials

Access Control(Continue...)

- Authentication and Authorization[1]



- Authentication Methods (Three-factor authentication) :
- Something you know (Something I know)
- Something you have (Something I Have)
- Something you are (Something I am)

[1] https://www.coursera.org/learn/cybersecurity/lecture/sQpu1/introduction-to-identity-credentials

Access Control(Continue...)

- Authentication and Authorization[1]

- Authentication Methods (Three-factor authentication) :
- Something you know (Something I know)
- Something you have (Something I Have)
- Something you are (Something I am)

[1] https://www.coursera.org/learn/cybersecurity/lecture/sQpu1/introduction-to-identity-credentials

Access Control(Continue...)

- Authentication and Authorization[1]

- Authentication Methods (Three-factor authentication) :
- Something you know (Something I know)
- Something you have (Something I Have)
- Something you are (Something I am)

[1] https://www.coursera.org/learn/cybersecurity/lecture/sQpu1/introduction-to-identity-credentials

Access Control(Continue...)

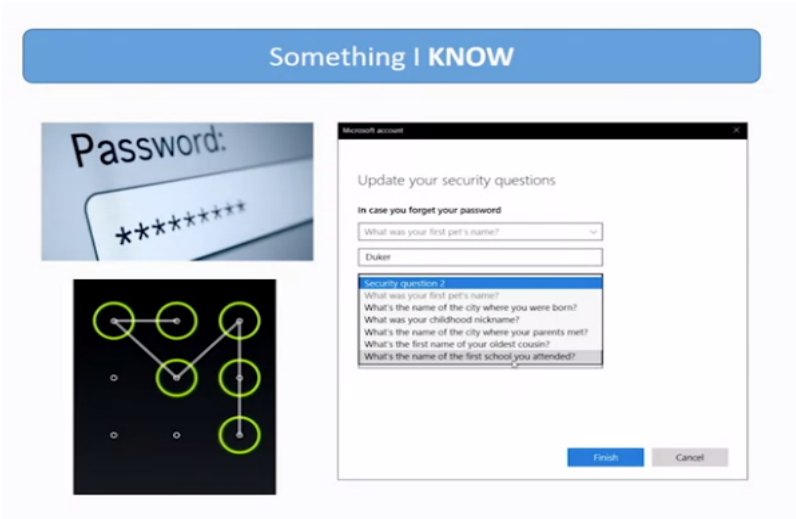
- Authentication and Authorization[1]

- Authentication Methods (Three-factor authentication) :
- Something you know (Something I know)
- Something you have (Something I Have)
- Something you are (Something I am)

[1] https://www.coursera.org/learn/cybersecurity/lecture/sQpu1/introduction-to-identity-credentials

Something you know

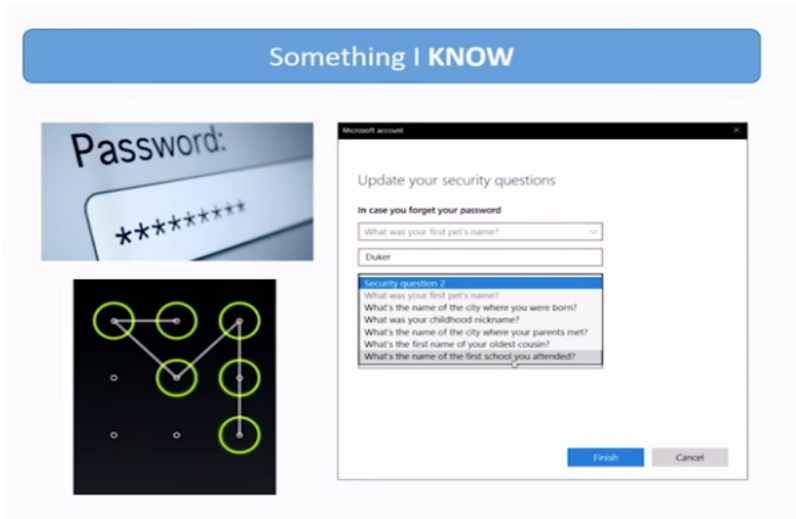
- Something you know (Something I know)



- Only user of system know
  -e.g. Password, Pin
  -knowledge of a secret distinguishes you from all other individuals
  -system simply needs to check to see if the person claiming to be you knows the secret
  -An attacker can last steal this kind of credential
- people will tend to choose passwords that are easy to remember, which usually means that the password is easy to guess

Something you know

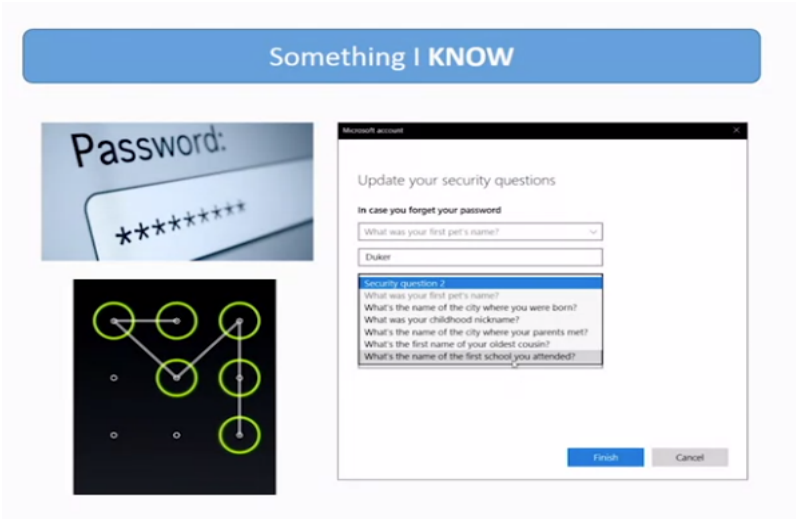
- Something you know (Something I know)



- Only user of system know
  -e.g. Password, Pin
  -knowledge of a secret distinguishes you from all other individuals
  -system simply needs to check to see if the person claiming to be you knows the secret
  -An attacker can last steal this kind of credential
- people will tend to choose passwords that are easy to remember, which usually means that the password is easy to guess

Something you know

- Something you know (Something I know)



- Only user of system know
  -e.g. Password, Pin
  -knowledge of a secret distinguishes you from all other individuals
  -system simply needs to check to see if the person claiming to be you knows the secret
  -An attacker can last steal this kind of credential
- people will tend to choose passwords that are easy to remember, which usually means that the password is easy to guess

Something you know

- Changing a password requires human intervention. Thus, compromised passwords could remain valid for longer than is desirable.
- And there must be some mechanism for resetting the password (because passwords will get forgotten and compromised).
- This mechanism could itself be vulnerable to social-engineering attacks, which rely on convincing a human with the authority to change or access information that it is necessary to do so.

Something you know

- Changing a password requires human intervention. Thus, compromised passwords could remain valid for longer than is desirable.
- And there must be some mechanism for resetting the password (because passwords will get forgotten and compromised).
- This mechanism could itself be vulnerable to social-engineering attacks, which rely on convincing a human with the authority to change or access information that it is necessary to do so.

Something you know

- Changing a password requires human intervention. Thus, compromised passwords could remain valid for longer than is desirable.
- And there must be some mechanism for resetting the password (because passwords will get forgotten and compromised).
- This mechanism could itself be vulnerable to social-engineering attacks, which rely on convincing a human with the authority to change or access information that it is necessary to do so.

Something you know

- There are three dimensions[2]
- Length
  - Longer passwords are better - A good way to get a long password that is seemingly random yet easy to remember
  - e.g. like the first words of a song and then generate the password from the first letters of the passphrase.
- Character set -more characters...greater the number of possible combinations of characters
  -so the larger the password space... require doing more work by an attacker.
- Randomness
  - Mathematically, it turns out that English has about 1.3 bits of information per character. Thus it takes 49 characters to get 64 bits of "secret", which comes out to about 10 words (at 5 characters on average per word).

[2] https://www.cs.cornell.edu/courses/cs513/2005fa/NNLauthPeople.html

Something you know

- There are three dimensions[2]
- Length
  - Longer passwords are better - A good way to get a long password that is seemingly random yet easy to remember
  - e.g. like the first words of a song and then generate the password from the first letters of the passphrase.
- Character set -more characters...greater the number of possible combinations of characters
  -so the larger the password space... require doing more work by an attacker.
- Randomness
  - Mathematically, it turns out that English has about 1.3 bits of information per character. Thus it takes 49 characters to get 64 bits of "secret", which comes out to about 10 words (at 5 characters on average per word).

[2] https://www.cs.cornell.edu/courses/cs513/2005fa/NNLauthPeople.html

Something you know

- There are three dimensions[2]
- Length
  - Longer passwords are better - A good way to get a long password that is seemingly random yet easy to remember
  - e.g. like the first words of a song and then generate the password from the first letters of the passphrase.
- Character set -more characters...greater the number of possible combinations of characters
  -so the larger the password space... require doing more work by an attacker.
- Randomness
  - Mathematically, it turns out that English has about 1.3 bits of information per character. Thus it takes 49 characters to get 64 bits of "secret", which comes out to about 10 words (at 5 characters on average per word).

[2] https://www.cs.cornell.edu/courses/cs513/2005fa/NNLauthPeople.html

Something you know

- There are three dimensions[2]
- Length
  - Longer passwords are better - A good way to get a long password that is seemingly random yet easy to remember
  - e.g. like the first words of a song and then generate the password from the first letters of the passphrase.
- Character set -more characters...greater the number of possible combinations of characters
  -so the larger the password space... require doing more work by an attacker.
- Randomness
  - Mathematically, it turns out that English has about 1.3 bits of information per character. Thus it takes 49 characters to get 64 bits of "secret", which comes out to about 10 words (at 5 characters on average per word).

[2] https://www.cs.cornell.edu/courses/cs513/2005fa/NNLauthPeople.html

Something you know : more on "Password"

- Password stored as HASH value
- possibility of offline dictionary attacks. - hash of every word in some dictionary - compares each hash with the stored password hashes. - if matched, the attacker has learned a password.
- Salt is a random number that is associated with a user and is added to that user's password when the hash is computed.
- system stores both h(password + salt) and the salt for each account
- Salt does not make it more difficult for an attacker to guess the password for a given account, since the salt for each account is stored in the clear. What salt does, however, is make it harder for the attacker to perpetrate an offline dictionary attack against all users. When salt is used, all the words in the dictionary would have to be rehashed for every user. What formerly could be seen as a "wholesale" attack has been transformed into a "retail" one.
- Salt is used in most UNIX implementations. The salt in early versions of UNIX was 12 bits, and it was formed from the system time and the process identifier when an account is created. Unfortunately, 12 bits is hopelessly small, nowadays. Even an old PC can perform 13,000 crypt/sec, which means such a PC so can hash a 20k word dictionary with every possible value of a 12 bit salt in 1 hour.
- Point to Explore : Secret Salt, Defense Against Password Theft: A Trusted Path [3]

[3] https://www.cs.cornell.edu/courses/cs513/2005fa/NNLauthPeople.html

Something you know : more on "Password"

- Password stored as HASH value
- possibility of offline dictionary attacks. - hash of every word in some dictionary - compares each hash with the stored password hashes. - if matched, the attacker has learned a password.
- Salt is a random number that is associated with a user and is added to that user's password when the hash is computed.
- system stores both h(password + salt) and the salt for each account
- Salt does not make it more difficult for an attacker to guess the password for a given account, since the salt for each account is stored in the clear. What salt does, however, is make it harder for the attacker to perpetrate an offline dictionary attack against all users. When salt is used, all the words in the dictionary would have to be rehashed for every user. What formerly could be seen as a "wholesale" attack has been transformed into a "retail" one.
- Salt is used in most UNIX implementations. The salt in early versions of UNIX was 12 bits, and it was formed from the system time and the process identifier when an account is created. Unfortunately, 12 bits is hopelessly small, nowadays. Even an old PC can perform 13,000 crypt/sec, which means such a PC so can hash a 20k word dictionary with every possible value of a 12 bit salt in 1 hour.
- Point to Explore : Secret Salt, Defense Against Password Theft: A Trusted Path [3]

[3] https://www.cs.cornell.edu/courses/cs513/2005fa/NNLauthPeople.html

Something you know : more on "Password"

- Password stored as HASH value
- possibility of offline dictionary attacks. - hash of every word in some dictionary - compares each hash with the stored password hashes. - if matched, the attacker has learned a password.
- Salt is a random number that is associated with a user and is added to that user's password when the hash is computed.
- system stores both h(password + salt) and the salt for each account
- Salt does not make it more difficult for an attacker to guess the password for a given account, since the salt for each account is stored in the clear. What salt does, however, is make it harder for the attacker to perpetrate an offline dictionary attack against all users. When salt is used, all the words in the dictionary would have to be rehashed for every user. What formerly could be seen as a "wholesale" attack has been transformed into a "retail" one.
- Salt is used in most UNIX implementations. The salt in early versions of UNIX was 12 bits, and it was formed from the system time and the process identifier when an account is created. Unfortunately, 12 bits is hopelessly small, nowadays. Even an old PC can perform 13,000 crypt/sec, which means such a PC so can hash a 20k word dictionary with every possible value of a 12 bit salt in 1 hour.
- Point to Explore : Secret Salt, Defense Against Password Theft: A Trusted Path [3]

[3] https://www.cs.cornell.edu/courses/cs513/2005fa/NNLauthPeople.html

Something you know : more on "Password"

- Password stored as HASH value
- possibility of offline dictionary attacks. - hash of every word in some dictionary - compares each hash with the stored password hashes. - if matched, the attacker has learned a password.
- Salt is a random number that is associated with a user and is added to that user's password when the hash is computed.
- system stores both h(password + salt) and the salt for each account
- Salt does not make it more difficult for an attacker to guess the password for a given account, since the salt for each account is stored in the clear. What salt does, however, is make it harder for the attacker to perpetrate an offline dictionary attack against all users. When salt is used, all the words in the dictionary would have to be rehashed for every user. What formerly could be seen as a "wholesale" attack has been transformed into a "retail" one.
- Salt is used in most UNIX implementations. The salt in early versions of UNIX was 12 bits, and it was formed from the system time and the process identifier when an account is created. Unfortunately, 12 bits is hopelessly small, nowadays. Even an old PC can perform 13,000 crypt/sec, which means such a PC so can hash a 20k word dictionary with every possible value of a 12 bit salt in 1 hour.
- Point to Explore : Secret Salt, Defense Against Password Theft: A Trusted Path [3]

[3] https://www.cs.cornell.edu/courses/cs513/2005fa/NNLauthPeople.html

Something you know : more on "Password"

- Password stored as HASH value
- possibility of offline dictionary attacks. - hash of every word in some dictionary - compares each hash with the stored password hashes. - if matched, the attacker has learned a password.
- Salt is a random number that is associated with a user and is added to that user's password when the hash is computed.
- system stores both h(password + salt) and the salt for each account
- Salt does not make it more difficult for an attacker to guess the password for a given account, since the salt for each account is stored in the clear. What salt does, however, is make it harder for the attacker to perpetrate an offline dictionary attack against all users. When salt is used, all the words in the dictionary would have to be rehashed for every user. What formerly could be seen as a "wholesale" attack has been transformed into a "retail" one.
- Salt is used in most UNIX implementations. The salt in early versions of UNIX was 12 bits, and it was formed from the system time and the process identifier when an account is created. Unfortunately, 12 bits is hopelessly small, nowadays. Even an old PC can perform 13,000 crypt/sec, which means such a PC so can hash a 20k word dictionary with every possible value of a 12 bit salt in 1 hour.
- Point to Explore : Secret Salt, Defense Against Password Theft: A Trusted Path [3]

[3] https://www.cs.cornell.edu/courses/cs513/2005fa/NNLauthPeople.html

Something you know : more on "Password"

- Password stored as HASH value
- possibility of offline dictionary attacks. - hash of every word in some dictionary - compares each hash with the stored password hashes. - if matched, the attacker has learned a password.
- Salt is a random number that is associated with a user and is added to that user's password when the hash is computed.
- system stores both h(password + salt) and the salt for each account
- Salt does not make it more difficult for an attacker to guess the password for a given account, since the salt for each account is stored in the clear. What salt does, however, is make it harder for the attacker to perpetrate an offline dictionary attack against all users. When salt is used, all the words in the dictionary would have to be rehashed for every user. What formerly could be seen as a "wholesale" attack has been transformed into a "retail" one.
- Salt is used in most UNIX implementations. The salt in early versions of UNIX was 12 bits, and it was formed from the system time and the process identifier when an account is created. Unfortunately, 12 bits is hopelessly small, nowadays. Even an old PC can perform 13,000 crypt/sec, which means such a PC so can hash a 20k word dictionary with every possible value of a 12 bit salt in 1 hour.
- Point to Explore : Secret Salt, Defense Against Password Theft: A Trusted Path [3]

[3] https://www.cs.cornell.edu/courses/cs513/2005fa/NNLauthPeople.html

Something you know : more on "Password"

- Password stored as HASH value
- possibility of offline dictionary attacks. - hash of every word in some dictionary - compares each hash with the stored password hashes. - if matched, the attacker has learned a password.
- Salt is a random number that is associated with a user and is added to that user's password when the hash is computed.
- system stores both h(password + salt) and the salt for each account
- Salt does not make it more difficult for an attacker to guess the password for a given account, since the salt for each account is stored in the clear. What salt does, however, is make it harder for the attacker to perpetrate an offline dictionary attack against all users. When salt is used, all the words in the dictionary would have to be rehashed for every user. What formerly could be seen as a "wholesale" attack has been transformed into a "retail" one.
- Salt is used in most UNIX implementations. The salt in early versions of UNIX was 12 bits, and it was formed from the system time and the process identifier when an account is created. Unfortunately, 12 bits is hopelessly small, nowadays. Even an old PC can perform 13,000 crypt/sec, which means such a PC so can hash a 20k word dictionary with every possible value of a 12 bit salt in 1 hour.
- Point to Explore : Secret Salt, Defense Against Password Theft: A Trusted Path [3]

[3] https://www.cs.cornell.edu/courses/cs513/2005fa/NNLauthPeople.html

Something you have (Something I Have)

- Something you have (Something I Have)

- e.g. Certificate, I-card, Smart Card

Something you have (Something I Have)

- Something you have (Something I Have)

- e.g. Certificate, I-card, Smart Card

Something you have (Something I Have)

- A magnetic strip card. (eg. Cornell ID, credit card) One serious problem with these cards is that they are fairly easy to duplicate. It only costs about $50 to buy a writer, and it's easy to get your hands on cards to copy them. To get around these problems, banks implement 2-factor authentication by requiring knowledge of a 4 to 7 character PIN whenever the card is used.
- Proximity card or RFID. These cards transmit stored information to a monitor via RF.
- Challenge/Response cards and Cryptographic Calculators. These are also called smart cards and perform some sort of cryptographic calculation. Sometimes the card will have memory, and sometimes it will have an associated PIN. A smart card transforms the authentication problem for humans, because we are no longer constrained by stringent computational and storage limitations. Unfortunately, today's smart cards are vulnerable to power-analysis attacks.
- One prevalent form of smartcard is the RSA secure id. It continuously displays encrypted time; and each RSA secure id encrypts with a different key.

Something you have (Something I Have)

- A magnetic strip card. (eg. Cornell ID, credit card) One serious problem with these cards is that they are fairly easy to duplicate. It only costs about $50 to buy a writer, and it's easy to get your hands on cards to copy them. To get around these problems, banks implement 2-factor authentication by requiring knowledge of a 4 to 7 character PIN whenever the card is used.
- Proximity card or RFID. These cards transmit stored information to a monitor via RF.
- Challenge/Response cards and Cryptographic Calculators. These are also called smart cards and perform some sort of cryptographic calculation. Sometimes the card will have memory, and sometimes it will have an associated PIN. A smart card transforms the authentication problem for humans, because we are no longer constrained by stringent computational and storage limitations. Unfortunately, today's smart cards are vulnerable to power-analysis attacks.
- One prevalent form of smartcard is the RSA secure id. It continuously displays encrypted time; and each RSA secure id encrypts with a different key.

Something you have (Something I Have)

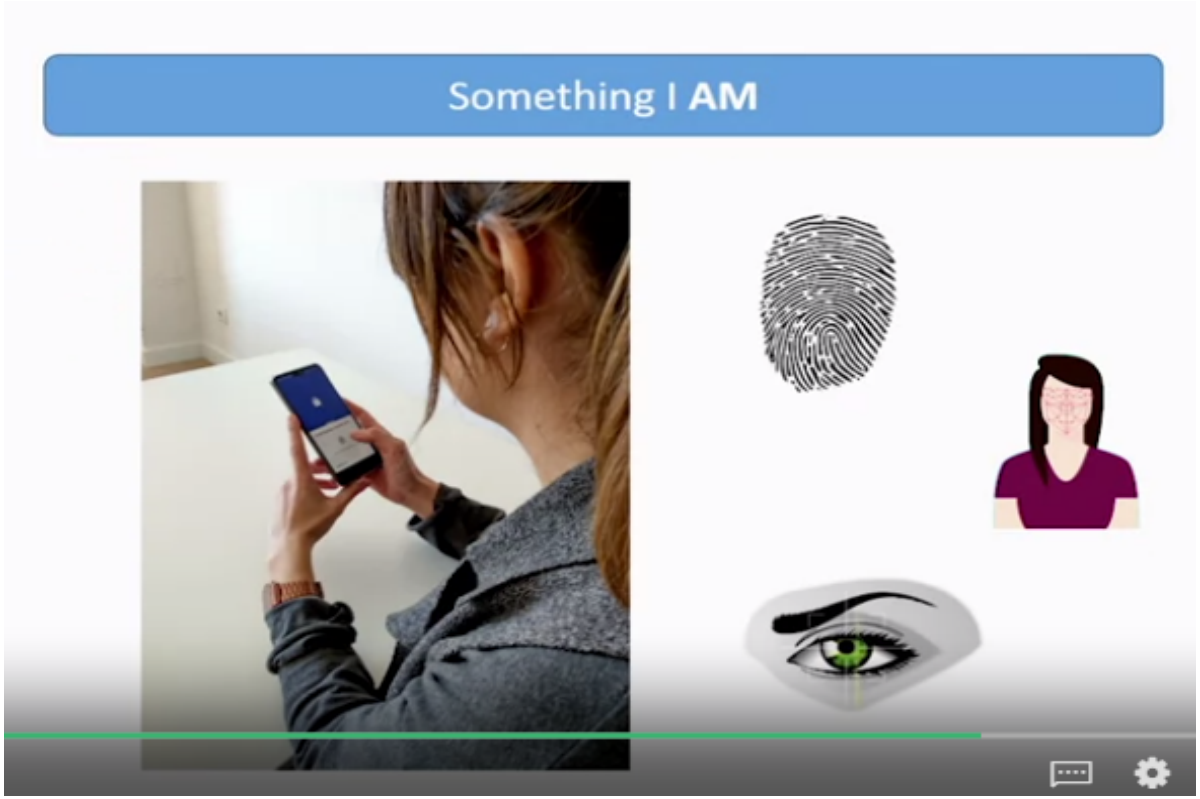
- A magnetic strip card. (eg. Cornell ID, credit card) One serious problem with these cards is that they are fairly easy to duplicate. It only costs about $50 to buy a writer, and it's easy to get your hands on cards to copy them. To get around these problems, banks implement 2-factor authentication by requiring knowledge of a 4 to 7 character PIN whenever the card is used.
- Proximity card or RFID. These cards transmit stored information to a monitor via RF.
- Challenge/Response cards and Cryptographic Calculators. These are also called smart cards and perform some sort of cryptographic calculation. Sometimes the card will have memory, and sometimes it will have an associated PIN. A smart card transforms the authentication problem for humans, because we are no longer constrained by stringent computational and storage limitations. Unfortunately, today's smart cards are vulnerable to power-analysis attacks.
- One prevalent form of smartcard is the RSA secure id. It continuously displays encrypted time; and each RSA secure id encrypts with a different key.

Something you have (Something I Have)

- A magnetic strip card. (eg. Cornell ID, credit card) One serious problem with these cards is that they are fairly easy to duplicate. It only costs about $50 to buy a writer, and it's easy to get your hands on cards to copy them. To get around these problems, banks implement 2-factor authentication by requiring knowledge of a 4 to 7 character PIN whenever the card is used.
- Proximity card or RFID. These cards transmit stored information to a monitor via RF.
- Challenge/Response cards and Cryptographic Calculators. These are also called smart cards and perform some sort of cryptographic calculation. Sometimes the card will have memory, and sometimes it will have an associated PIN. A smart card transforms the authentication problem for humans, because we are no longer constrained by stringent computational and storage limitations. Unfortunately, today's smart cards are vulnerable to power-analysis attacks.
- One prevalent form of smartcard is the RSA secure id. It continuously displays encrypted time; and each RSA secure id encrypts with a different key.

Something you are (Something I am)

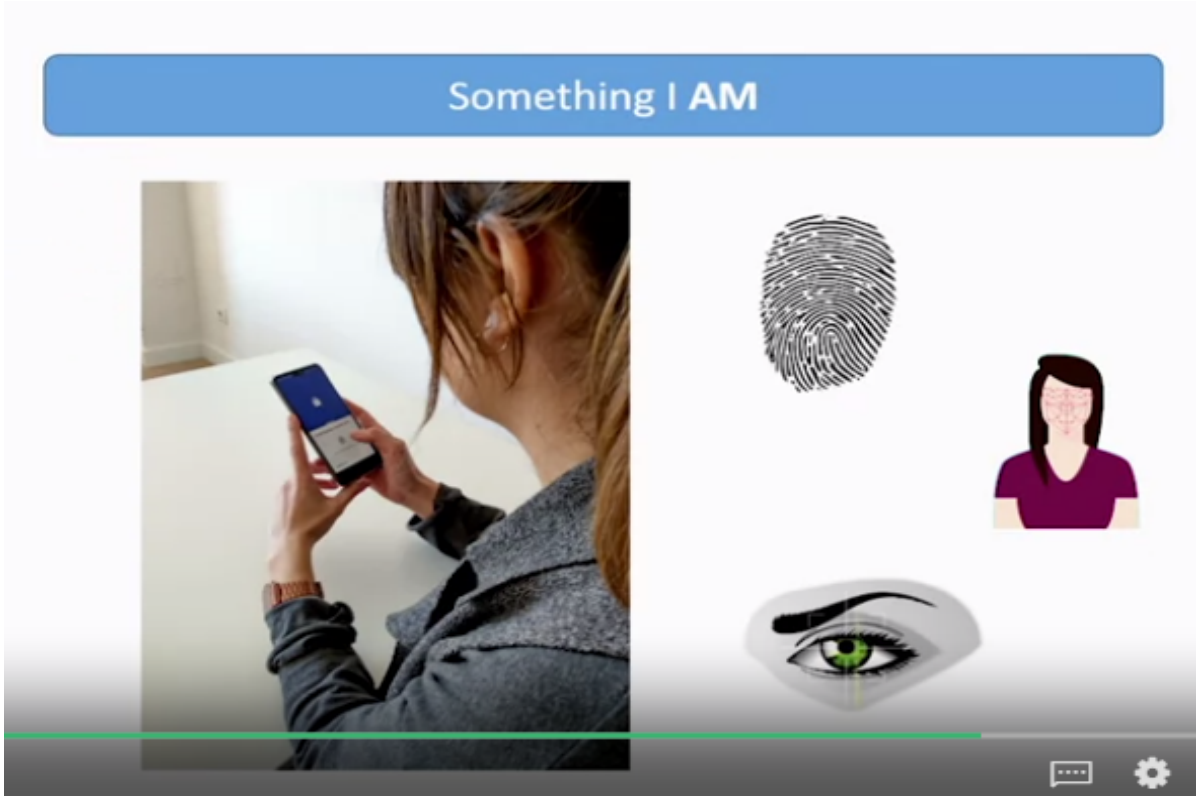
- Something you are (Something I am)



- e.g. Certificate, I-card, Smart Card

Something you are (Something I am)

- Something you are (Something I am)



- e.g. Certificate, I-card, Smart Card

Something you are (Something I am)

- employ behavioral and physiological characteristics
- we might use Retinal scan, Fingerprint reader, Handprint reader, Voice print, Keystroke timing ,Signature
- with some agreed error rate (For example, fingerprint readers today normally exhibit error rates upwards of 5%.)
- Methods to subvert a fingerprint reader give some indication of the difficulties of deploying unsupervised biometric sensors as the sole means of authenticating humans.
  -Steal a finger
  -Steal a fingerprint
  -Replace the biometric sensor

Something you are (Something I am)

- employ behavioral and physiological characteristics
- we might use Retinal scan, Fingerprint reader, Handprint reader, Voice print, Keystroke timing ,Signature
- with some agreed error rate (For example, fingerprint readers today normally exhibit error rates upwards of 5%.)
- Methods to subvert a fingerprint reader give some indication of the difficulties of deploying unsupervised biometric sensors as the sole means of authenticating humans.
  -Steal a finger
  -Steal a fingerprint
  -Replace the biometric sensor

Something you are (Something I am)

- employ behavioral and physiological characteristics
- we might use Retinal scan, Fingerprint reader, Handprint reader, Voice print, Keystroke timing ,Signature
- with some agreed error rate (For example, fingerprint readers today normally exhibit error rates upwards of 5%.)
- Methods to subvert a fingerprint reader give some indication of the difficulties of deploying unsupervised biometric sensors as the sole means of authenticating humans.
  -Steal a finger
  -Steal a fingerprint
  -Replace the biometric sensor

Something you are (Something I am)

- employ behavioral and physiological characteristics
- we might use Retinal scan, Fingerprint reader, Handprint reader, Voice print, Keystroke timing ,Signature
- with some agreed error rate (For example, fingerprint readers today normally exhibit error rates upwards of 5%.)
- Methods to subvert a fingerprint reader give some indication of the difficulties of deploying unsupervised biometric sensors as the sole means of authenticating humans.
  -Steal a finger
  -Steal a fingerprint
  -Replace the biometric sensor

Something you are (Something I am)

- There are several well known problems with biometric-based authentication schemes:
  -Reliability of the method
  -Cost and availability
  -Unwillingness or inability to interact with biometric input devices
  -Compromise the biometric database or system

Research Problem 2

- Post 9/11, the United States started searching airline passengers for bombs and other potentially dangerous material. More recently, in reaction to the Madrid and London subway bombings, other governments have started searching subway customers.
  In all of these schemes, the cost of searching a suspect is significant and it is too costly to search every passenger. Some sort of sampling is then employed, which leads to a design choice about who gets selected for searches:
  Select randomly among all passengers.
  Select randomly among passengers satisfying certain predefined profiles.
  Adopt (1) and you end up searching babies, grandmothers, and congressman; adopt (2) and you might only search males of a certain age and ethnicity.
  Given a fixed budget for performing searches, which of (1) and (2) is likely to be more effective at decreasing the chances of successful future terrorist attacks on airplanes and/or subways.

Authentication Protocols

- Different Categories of Authentication Protocols:
  - Reciprocity of Authentication (Mutual/One Way)
  - Type of Cryptography (Symmetric/Asymmetric)
  - Key Exchange
  - Computational and Communication Efficiency
  - Real time involvement of Third Party (online vs offline)

Authentication Protocols

- Different Categories of Authentication Protocols:
- Reciprocity of Authentication (Mutual/One Way)
- Type of Cryptography (Symmetric/Asymmetric)
- Key Exchange
- Computational and Communication Efficiency
- Real time involvement of Third Party (online vs offline)

Authentication Protocols

- Different Categories of Authentication Protocols:
- Reciprocity of Authentication (Mutual/One Way)
- Type of Cryptography (Symmetric/Asymmetric)
- Key Exchange
- Computational and Communication Efficiency
- Real time involvement of Third Party (online vs offline)

Authentication Protocols

- Different Categories of Authentication Protocols:
- Reciprocity of Authentication (Mutual/One Way)
- Type of Cryptography (Symmetric/Asymmetric)
- Key Exchange
- Computational and Communication Efficiency
- Real time involvement of Third Party (online vs offline)

Authentication Protocols

- Different Categories of Authentication Protocols:
- Reciprocity of Authentication (Mutual/One Way)
- Type of Cryptography (Symmetric/Asymmetric)
- Key Exchange
- Computational and Communication Efficiency
- Real time involvement of Third Party (online vs offline)

Authentication Protocols

- Different Categories of Authentication Protocols:
- Reciprocity of Authentication (Mutual/One Way)
- Type of Cryptography (Symmetric/Asymmetric)
- Key Exchange
- Computational and Communication Efficiency
- Real time involvement of Third Party (online vs offline)