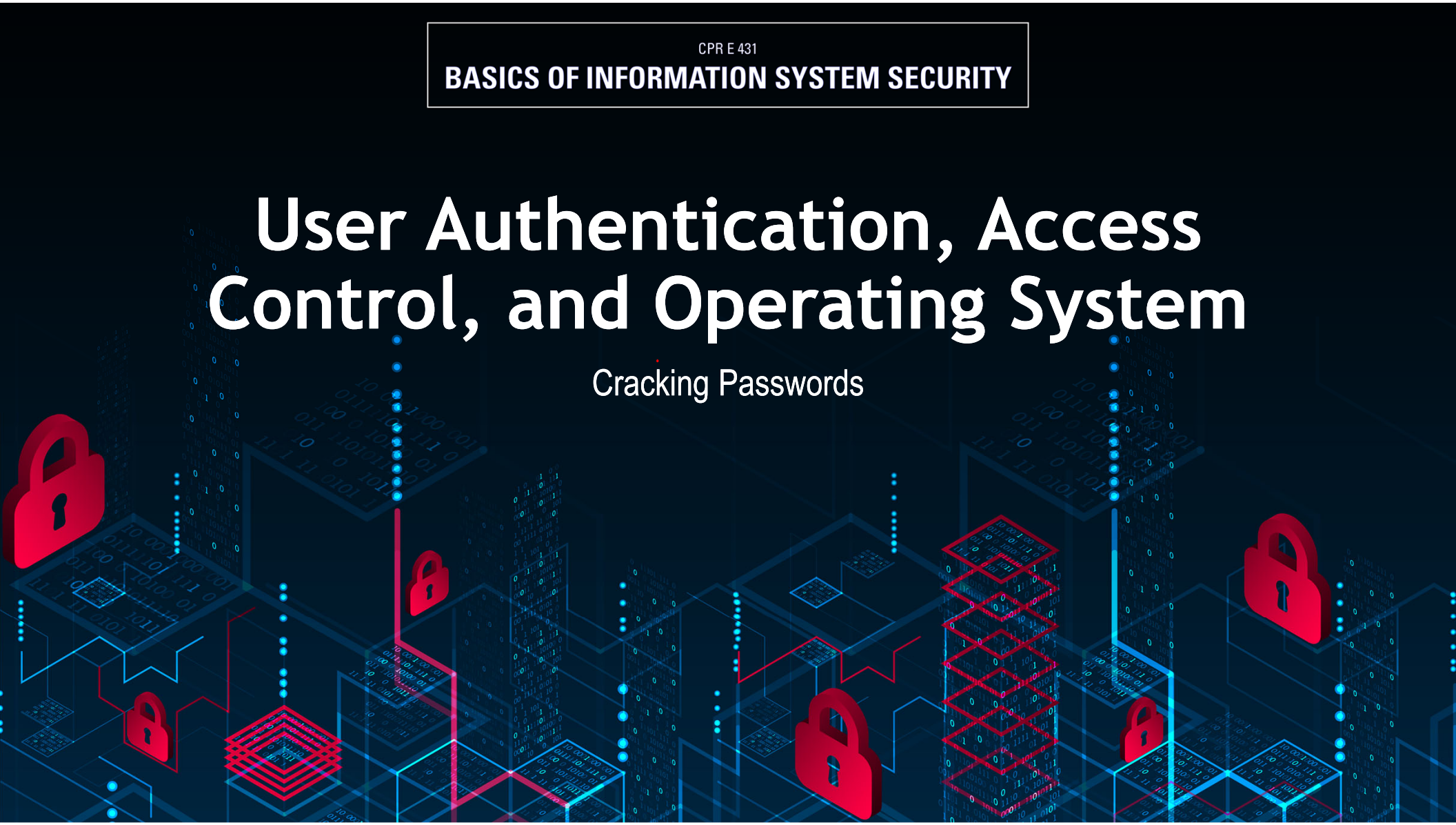
CPR E 431

BASICS OF INFORMATION SYSTEM SECURITY

User Authentication, Access Control, and Operating System

Cracking Passwords



Video Summary

- Brute Force Attack on Hashed Passwords
- Hashing Speed
- Preventing Hashing Attacks
- Rainbow Tables

Brute Force Attack on Hashed Passwords

- Aim: given one (or more) target hash value, find the original password
- Start with large set of possible passwords (e.g. from dictionary, all possible n-character combinations)
- Calculate hash of possible password, compare with target hash
  - if match, original password is found
  - else, try next possible password
- Attack duration depends on size of possible password set

Hashing the Passwords

username	H(password)
john	06c219e5bc8378f3a8a3f83b4b7e4649
sandy	5fc2bb44573c7736badc8382b43fbeae
daniel	06c219e5bc8378f3a8a3f83b4b7e4649
...	...
steve	75127c78fd791c3f92a086c59c71ece0

- Brute force on n-bit hash value: 2^n attempts
- For MD5 128 bit: 2^{128} attempts (how long does this take?)
- How many hashes your computer calculate per second?

4×10^6 hashes/sec

Hashing the Passwords

username	H(password)
john	06c219e5bc8378f3a8a3f83b4b7e4649
sandy	5fc2bb44573c7736badc8382b43fbeae
daniel	06c219e5bc8378f3a8a3f83b4b7e4649
...	...
steve	75127c78fd791c3f92a086c59c71ece0

- Brute force on n-bit hash value: 2^n attempts
- For MD5 128 bit: 2^{128} attempts (how long does this take?)
- How many hashes your computer calculate per second?
- @ 4×10^6 hashes/sec

$2^{128} / (4 \times 10^6 * 60 * 60 * 24) = 9.85 \times 10^{26}$ days

Hashing the Passwords

username	H(password)
john	06c219e5bc8378f3a8a3f83b4b7e4649
sandy	5fc2bb44573c7736badc8382b43fbeae
daniel	06c219e5bc8378f3a8a3f83b4b7e4649
...	...
steve	75127c78fd791c3f92a086c59c71ece0

- Brute force on n-bit hash value: 2^n attempts
- For MD5 128 bit: 2^{128} attempts (how long does this take?)
- How many hashes your computer calculate per second?
- @ $4x10^6$ hashes/sec
- How long ➔ 2^{128} /(4000000*60*60*24)=$9.85x10^{26}$ days = $2.7x10^{24}$ years

Hashing the Passwords

- What if we used a GPU? (gaming computers or mining hardware)
- 10^6 hashes/sec: still TOO LONG

Hashing the Passwords

- What if we used a GPU? (gaming computers or mining hardware)
- 10^6 hashes/sec: still TOO LONG
- What about using GPU and parallel computing?➔ 10^{10} hashes/sec
- How many passwords the user can choose from given that you have a maximum of 8 characters?

$1 \rightarrow 94$

$2 \rightarrow (94)^2$

$3 \rightarrow (94)^3$

$8 \rightarrow (94)^8$

Hashing the Passwords

- Worst case: $94^8 + 94^7 + 94^6 + 94^5 + 94^4 + 94^3 + 94^2 + 94^1 = 6.16 \times 10^{15}$ possible passwords

$$94^8 \simeq 6.16 \times 10^{15}$$

- If we are used a GPU (10^{10} hashes/sec).. How long it will take us to calculate the hashes of all passwords?

$$6.16 \times 10^{15} / (10^{10} \times 60 \times 60 \times 24)$$

7 days

Hashing the Passwords

- Worst case: $94^8 + 94^7 + 94^6 + 94^5 + 94^4 + 94^3 + 94^2 + 94^1 = 6.16x10^{15}$ possible passwords
- If we are used a GPU (10^{10} hashes/sec).. How long it will take us to calculate the hashes of all passwords?
- $6.16x10^{15} / (10^{10} *60*60*24) = 7$ days!!
- How to prevent such an attach?

Hashing the Passwords

- Worst case: $94^8 + 94^7 + 94^6 + 94^5 + 94^4 + 94^3 + 94^2 + 94^1 = 6.16 \times 10^{15}$ possible passwords
- If we are used a GPU (10^{10} hashes/sec).. How long it will take us to calculate the hashes of all passwords?
- $6.16 \times 10^{15} / (10^{10} * 60 * 60 * 24) = 7$ days!!
- How to prevent such an attach? ➔ Use a slower hash function

Hash Type	PC1	PC2
MD5	8581 Mh/s	2753 Mh/s
SHA1	3037 Mh/s	655 Mh/s
SHA256	1122 Mh/s	355 Mh/s
SHA512	414 Mh/s	104 Mh/s
SHA-3(Keccak)	179 Mh/s	92 Mh/s

Hashing the Passwords

- Worst case: 94^8 $94^7 + 94^6 + 94^5 + 94^4 + 94^3 + 94^2 + 94^1 = 6.16x10^{15}$ possible passwords
- If we are used a GPU (10^{10} hashes/sec).. How long it will take us to calculate the hashes of all passwords?
- $6.16x10^{15} / (10^{10} *60*60*24) = 7$ days!!
- How to prevent such an attach?
  - ✓ More characters in the password (for example 9 digits)

$(94)^9$

Cracking Passwords

- Store passwords and hash values in advance (instead of generating them)
- The question is how big is it?

  Password is 8 Bytes + hash is 128 bits (if using MD5)

  (8 Byte + 16 Byte) x 94^8 = 1.4x1017 Bytes = 146 TB (approx.)

- Instead of generating this huge amount of data we can use

Rainbow Tables

Cracking Passwords

➢Rainbow Tables

MD5 Rainbow Tables

Table ID	Charset	Plaintext Length	Key Space	Success Rate	Table Size
md5_ascii-32-95#1-7	ascii-32-95	1 to 7	70,576,641,626,495	99.9 %	52 GB 64 GB
md5_ascii-32-95#1-8	ascii-32-95	1 to 8	6,704,780,954,517,120	96.8 %	460 GB 576 GB
md5_mixalpha-numeric#1-8	mixalpha-numeric	1 to 8	221,919,451,578,090	99.9 %	127 GB 160 GB
md5_mixalpha-numeric#1-9	mixalpha-numeric	1 to 9	13,759,005,997,841,642	96.8 %	690 GB 864 GB

- Lookup on 0.5 TB Rainbow Table will take only hours to find the password

http://project-rainbowcrack.com/table.htm

Pre-calculated Hashes & Rainbow Tables

- How big is such a database of pre-calculated hashes?
  - In raw form, generally too big to be practical (100's, 1000's of TB)
  - Using specialised data structures (e.g. Rainbow tables), can obtain manageable size, e.g. 1 TB
- Trade-off: reduce search time, but increase storage space
- Countermeasures:
  - Longer passwords
  - Slower hash algorithms
  - Salting the password before hashing

Video Summary

- Brute Force Attack on Hashed Passwords
- Hashing Speed
- Preventing Hashing Attacks
- Rainbow Tables