SSCP$^{Q\&As}$

System Security Certified Practitioner (SSCP)

Pass ISC SSCP Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.passapply.com/sscp.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by ISC Official Exam Center

QUESTION 1

Which of the following is not a preventive login control?

A. Last login message

B. Password aging

C. Minimum password length

D. Account expiration

Correct Answer: A

The last login message displays the last login date and time, allowing a user to discover if their account was used by someone else. Hence, this is rather a detective control.

Source: RUSSEL, Deborah and GANGEMI, G.T. Sr., Computer Security Basics, O\\'Reilly, July 1992 (page 63).

QUESTION 2

A contingency plan should address:

A. Potential risks.

B. Residual risks.

C. Identified risks.

D. All answers are correct.

Correct Answer: D

Because it is rarely possible or cost effective to eliminate all risks, an attempt is made to reduce risks to an acceptable level through the risk assessment process. This process allows, from a set of potential risks (whether likely or not), to come up with a set of identified, possible risks.

The implementation of security controls allows reducing the identified risks to a smaller set of residual risks. Because these residual risks represent the complete set of situations that could affect system performance, the scope of the contingency plan may be reduced to address only this decreased risk set. As a result, the contingency plan can be narrowly focused, conserving resources while ensuring an effective system recovery capability.

Source: SWANSON, Marianne, and al., National Institute of Standards and Technology (NIST), NIST Special Publication 800-34, Contingency Planning Guide for Information Technology Systems, December 2001 (page 7).

QUESTION 3

Which of the following ASYMMETRIC encryption algorithms is based on the difficulty of FACTORING LARGE NUMBERS?

A. El Gamal

B. Elliptic Curve Cryptosystems (ECCs)

C. RSA

D. International Data Encryption Algorithm (IDEA)

Correct Answer: C

Named after its inventors Ron Rivest , Adi Shamir and Leonard Adleman is based on the difficulty of factoring large prime numbers.

Factoring a number means representing it as the product of prime numbers. Prime numbers, such as 2, 3, 5, 7, 11, and 13, are those numbers that are not evenly divisible by any smaller number, except 1. A non-prime, or composite number, can be written as the product of smaller primes, known as its prime factors. 665, for example is the product of the primes 5, 7, and 19. A number is said to be factored when all of its prime factors are identified. As the size of the number increases, the difficulty of factoring increases rapidly.

The other answers are incorrect because:

El Gamal is based on the discrete logarithms in a finite field.

Elliptic Curve Cryptosystems (ECCs) computes discrete logarithms of elliptic curves.

International Data Encryption Algorithm (IDEA) is a block cipher and operates on 64 bit blocks of data and is a SYMMETRIC algorithm.

Reference : Shon Harris , AIO v3 , Chapter-8 : Cryptography , Page : 638

QUESTION 4

Which of the following is NOT a common category/classification of threat to an IT system?

A. Human

B. Natural

C. Technological

D. Hackers

Correct Answer: D

Hackers are classified as a human threat and not a classification by itself.

All the other answers are incorrect. Threats result from a variety of factors, although they are classified in three types: Natural (e.g., hurricane, tornado, flood and fire), human (e.g. operator error, sabotage, malicious code) or technological (e.g. equipment failure, software error, telecommunications network outage, electric power failure).

Reference:

SWANSON, Marianne, and al., National Institute of Standards and Technology (NIST), http://csrc.nist.gov/publications/nistpubs/800-34-rev1/sp800-34-rev1_errata-Nov11-2010.pdf, June 2002 (page 6).

QUESTION 5

The primary purpose for using one-way hashing of user passwords within a password file is which of the following?

A. It prevents an unauthorized person from trying multiple passwords in one logon attempt.

B. It prevents an unauthorized person from reading the password.

C. It minimizes the amount of storage required for user passwords.

D. It minimizes the amount of processing time used for encrypting passwords.

Correct Answer: B

The whole idea behind a one-way hash is that it should be just that - one-way. In other words, an attacker should not be able to figure out your password from the hashed version of that password in any mathematically feasible way (or within any reasonable length of time).

Password Hashing and Encryption

In most situations , if an attacker sniffs your password from the network wire, she still has some work to do before she actually knows your password value because most systems hash the password with a hashing algorithm, commonly MD4 or MD5, to ensure passwords are not sent in cleartext.

Although some people think the world is run by Microsoft, other types of operating systems are out there, such as Unix and Linux. These systems do not use registries and SAM databases, but contain their user passwords in a file cleverly called "shadow." Now, this shadow file does not contain passwords in cleartext; instead, your password is run through a hashing algorithm, and the resulting value is stored in this file.

Unixtype systems zest things up by using salts in this process. Salts are random values added to the encryption process to add more complexity and randomness. The more randomness entered into the encryption process, the harder it is for the bad guy to decrypt and uncover your password. The use of a salt means that the same password can be encrypted into several thousand different formats. This makes it much more difficult for an attacker to uncover the right format for your system.

Password Cracking tools

Note that the use of one-way hashes for passwords does not prevent password crackers from guessing passwords. A password cracker runs a plain-text string through the same one-way hash algorithm used by the system to generate a hash, then compares that generated has with the one stored on the system. If they match, the password cracker has guessed your password. This is very much the same process used to authenticate you to a system via a password. When you type your username and password, the system hashes the password you typed and compares that generated hash against the one stored on the system if they match, you are authenticated.

Pre-Computed password tables exists today and they allow you to crack passwords on Lan Manager (LM) within a VERY short period of time through the use of Rainbow Tables. A Rainbow Table is a precomputed table for reversing cryptographic hash functions, usually for cracking password hashes. Tables are usually used in recovering a plaintext password up to a certain length consisting of a limited set of characters. It is a practical example of a space/time trade-off also called a Time-Memory trade off, using more computer processing time at the cost of less storage when calculating a hash on every attempt, or less processing time and more storage when compared to a simple lookup table with one entry per hash. Use of a key derivation function that employs a salt makes this attack unfeasible.

You may want to review "Rainbow Tables" at the links:

http://en.wikipedia.org/wiki/Rainbow_table

http://www.antsight.com/zsl/rainbowcrack/

Today\\'s password crackers:

Meet oclHashcat. They are GPGPU-based multi-hash cracker using a brute-force attack (implemented as mask attack), combinator attack, dictionary attack, hybrid attack, mask attack, and rule-based attack.

This GPU cracker is a fusioned version of oclHashcat-plus and oclHashcat-lite, both very well- known suites at that time, but now deprecated. There also existed a now very old oclHashcat GPU cracker that was replaced w/ plus and lite, which - as said - were then merged into oclHashcat 1.00 again.

This cracker can crack Hashes of NTLM Version 2 up to 8 characters in less than a few hours. It is definitively a game changer. It can try hundreds of billions of tries per seconds on a very large cluster of GPU\\'s. It supports up to 128 Video Cards at once. I am stuck using Password what can I do to better protect myself?

You could look at safer alternative such as Bcrypt, PBKDF2, and Scrypt.

bcrypt is a key derivation function for passwords designed by Niels Provos and David Mazières, based on the Blowfish cipher, and presented at USENIX in 1999. Besides incorporating a salt to protect against rainbow table attacks, bcrypt is an adaptive function: over time, the iteration count can be increased to make it slower, so it remains resistant to brute-force search attacks even with increasing computation power.

In cryptography, scrypt is a password-based key derivation function created by Colin Percival, originally for the Tarsnap online backup service. The algorithm was specifically designed to make it costly to perform large-scale custom hardware attacks by requiring large amounts of memory. In 2012, the scrypt algorithm was published by the IETF as an Internet Draft, intended to become an informational RFC, which has since expired. A simplified version of scrypt is used as a proof-of-work scheme by a number of cryptocurrencies, such as Litecoin and Dogecoin.

PBKDF2 (Password-Based Key Derivation Function 2) is a key derivation function that is part of RSA Laboratories\\' Public-Key Cryptography Standards (PKCS) series, specifically PKCS #5 v2.0, also published as Internet Engineering Task Force\\'s RFC 2898. It replaces an earlier standard, PBKDF1, which could only produce derived keys up to 160 bits long.

PBKDF2 applies a pseudorandom function, such as a cryptographic hash, cipher, or HMAC to the input password or passphrase along with a salt value and repeats the process many times to produce a derived key, which can then be used as a cryptographic key in subsequent operations. The added computational work makes password cracking much more difficult, and is known as key stretching. When the standard was written in 2000, the recommended minimum number of iterations was 1000, but the parameter is intended to be increased over time as CPU speeds increase. Having a salt added to the password reduces the ability to use precomputed hashes (rainbow tables) for attacks, and means that multiple passwords have to be tested individually, not all at once. The standard recommends a salt length of at least 64 bits. The other answers are incorrect:

"It prevents an unauthorized person from trying multiple passwords in one logon attempt." is incorrect because the fact that a password has been hashed does not prevent this type of brute force password guessing attempt.

"It minimizes the amount of storage required for user passwords" is incorrect because hash algorithms always generate the same number of bits, regardless of the length of the input. Therefore, even short passwords will still result in a longer hash and not minimize storage requirements.

"It minimizes the amount of processing time used for encrypting passwords" is incorrect because the processing time to encrypt a password would be basically the same required to produce a one-way has of the same password.

Reference(s) used for this question:

http://en.wikipedia.org/wiki/PBKDF2

http://en.wikipedia.org/wiki/Scrypt

http://en.wikipedia.org/wiki/Bcrypt

Harris, Shon (2012-10-18). CISSP All-in-One uide, 6th Edition (p. 195) . McGraw-Hill. Kindle Edition.