CompTIA Security + 6.0 Cryptography and PKI

Filename: comptia-secplussy0501-6-1-basic_concepts_of_cryptography
Title: Basic Concepts of Cryptography
Subtitle: CompTIA Security+ (SY0-501)

6.1 Basic Concepts of Cryptography

- 6.1 Compare and contrast basic concepts of cryptography
  - Symmetric algorithms
    - Basic Diagram
  - Modes of operation
    - Basic Diagram
  - Asymmetric algorithms
  - Hashing
  - Salt, IV, nonce
    - Salts
      - Adds additional bits to a password prior to a hashing operation being performed on it
    - IVs
      - added to plaintext to ensure that identical plaintexts inputs do not produce the same ciphertext outputs
      - http://aesencryption.net/
      - Plaintext - This is my secret text
      - Key - QF0yS%7#LQD6'7hq]{Z94k/3Y'kAw4
      - Cipher Text - uIIXiWX2uowfJwaJV+IBU7bVO5oG6n7ez2VdrpkXrCA=
  - Elliptic curve
    - Smaller key sizes
    - Stronger Keys in smaller sizes
    - ECC 256 is as strong as thousands of times stronger than a RSA key
  - Weak/deprecated algorithms
  - Key exchange
    - Mention Diffie-Hellman
    - Will be demonstrated in PFS below
  - Digital signatures
    - See Diagram
  - Diffusion
    - A principle in en encryption where a modification of a single bit of plaintext should modify a large number of bits in the ciphertext(goal is about 50%).
  - Confusion
    - Seeks to make the relationship between the key and the ciphertext as complex and involved as much as possible.
  - Collision
  - Steganography
    - Steganography lets you send messages without raising suspicion, but runs the risk of being discovered
    - Encryption lets you conceal the message but not the fact that the message exists
  - Obfuscation
    - Code obfuscation makes it harder to reverse engineer, if a company is worried about that
    - If you obfuscate the licensing process of the software it makes is more difficult to reverse engineer that process
  - Stream vs. block
    - Review
  - Key strength
  - Session keys
    - SSL/TLS
    - Mentioned in PFS below
  - Ephemeral key
    - A cryptographic key that is generated for each execution of a key establishment process
    - A unique key every time a key is established
    - Ephemeral is lasting a short time, short lifecycle, short lived
  - Secret algorithm
  - Data-in-transit
    - Data that is traversing a network
    - Data that is in a buffer waiting to be transmitted/processed
    - Data that is in system memory waiting to be processed
    - Data is protected with ACLs, encryption and hashing
  - Data-at-rest
    - Data stored on a device or storage media
    - Data that is not being used by applications
    - Data that is not being transfer over network medium
    - Examples
      - Backups
      - Offsite Backups
      - External media

        - Data is proteced with encryption, hashing and ACLs
    - Data-in-use
        - Data that is actively being processed by applications
        - Data that is being viewed and/or modified by a user
        - Data is protected with ACLs
    - Random/pseudo-random number generation
        - TRNG
            - TRNG produces a random string based on a physical process like static in airwaves, ocean waves, thermal noise
        - PRNG
            - PRNG produce a random string of numbers via a mathematical algorithm
            - Not truly random, however when the number string is compared to a truly random string it APPEARs random
        - Hardware-based
            - a device that generated random number strings
            - ChaosKey http://altusmetrum.org/ChaosKey/
            - Araneus Alea II http://bit.ly/2qemY89
            - PureQuantum™ http://bit.ly/2p3PknC
        - Software-based
            - https://www.random.org/
    - Key stretching
    - Implementation vs. algorithm selection
        - Crypto service provider
            - HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\Defaults\Provider
            - Software Library with individual modules
            - They implement Microsoft's CryptoAPI
            - The CSPs implement the algorithms and standards
            - CSPs abstract the cryptographic components so application do not have to
            - CSP > CryptoAPI > Application
            - Implementation is in the form of a dll
        - Crypto modules
            - A hardware device or software that performs cryptographic operations within a physical or logical boundary
            - Defined in FIPS 140-2 (Federal Information Processing Standards): Security Requirements for Cryptographic Modules
    - Perfect forward secrecy
        - See slide
    - Security through obscurity
    - Common use cases
        - Low power devices(symmetric)
        - Low latency(symmetric, smaller keys)
        - High resiliency
        - Supporting confidentiality(asymmetric, symmetric)
        - Supporting integrity(digital certificates, hashing)
        - Supporting obfuscation (XOR process, ROT13, substitution ciphers)
        - Supporting authentication (asymmetric encryption, perfect forward secrecy, key stretching)
        - Supporting non-repudiation(digital signatures, asymmetric, Perfect Forward Secrecy)
        - Resource vs. security constraints
            - Security constraints define the level of privileges to a collection of resources
            - Security constraints grant or deny access to a resource