Cybersecurity

Architecture and Design

2.8.8 Cryptography Limitations

What are some important questions to ask when determining cryptographic keys?

Overview
The student will summarize the basics of cryptographic concepts.

Grade Level(s)
10, 11, 12

Cyber Connections

- Threats & Vulnerabilities
- Networks & Internet
- Hardware & Software

This content is based upon work supported by the US Department of Homeland Security's Cybersecurity & Infrastructure Security Agency under the Cybersecurity Education Training and Assistance Program (CETAP).

Teacher Notes:

CompTIA SY0-601 Security+ Objectives

Objective 2.8

- Summarize the basics of cryptographic concepts.
    - Limitations
        - Speed
        - Size
        - Weak keys
        - Time
        - Longevity
        - Predictability
        - Reuse
        - Entropy
        - Computational overheads
        - Resource vs. security constraints

Cryptography Limitations

Push it to the Limit!

As computers advance, the need for more complex cryptographic techniques becomes necessary. What was originally thought secure, like a Vigenère Cipher, becomes nearly trivial with ever advancing computers. There will be a time when our 64-bit computers are "jokes" in comparison to the current technology of that age. Regardless of when and what the technology is, there are certain limitations that remain constant.

Speed, size, and time go hand-in-hand with regard to cryptography. If an encryption/decryption is too large, it will take too long to communicate back and forth. With email, that is not a huge issue, but if you need instant communication, this can be a real problem. Some programs will even "time out" if it takes too long for certain computations.

There are also scenarios where someone uses a good cryptographic algorithm, like AES, but implements a weak key. An obvious example is WEP, which uses an improper implementation of the RC4 encryption algorithm.

The longevity of a cryptographic key needs to be considered as well. Important questions to ask are "How long is this key expected to remain secure?", "Once compromised, what's the follow-up?", "How do we properly 'dispose' of a key?".

Teacher Notes:

Predictability is another concern. What if you are so infatuated with a sports team that every key you use/generate involves that team? If a user or organization has a known pattern for generating keys, it can be easier for a malicious user to exploit that pattern to generate false messages or crack the key.

As obvious as this may sound, do not reuse keys! Impersonation, man-in-the-middle, and passive decryption attacks are possible when keys are reused. In cryptography, entropy refers to the randomness collected by a system for use in algorithms that require random data (it's not fully random, rather pseudo-random). Without proper randomness, malicious users could crack the key easily.

The final few items that always need to be considered are computational overheads and resource vs. security constraints. Computers can only handle so much information, so if a key is too complicated, we will have the same issues with time, speed, and size. There are also issues with power consumption and heat generation, which result in higher costs. A balance is needed between proper security and resource exhaustion.