CS2107 Tutorial 1 (Introduction & Encryption)

School of Computing, NUS

23–27 August 2021

1. Alice was the Web administrator of the company *WhatSecurity**. A malicious attacker sent an email to Alice. The email instructed Alice to click on a link so as to login to the company's HR system to view a report. In the email, information of the "sender" had been modified to be the HR manager of *WhatSecurity*. Alice wrongly believed that the email was indeed sent by the manager, and followed the instructions. In doing so, she revealed her password to the attacker. Using Alice's password, the attacker then logged-in to the Web server, and invoked many processes. As a result, the server got overloaded.

   With respect to the security requirements mentioned in the lecture (confidentiality, integrity, availability, authenticity, etc.), discuss what aspects of security were compromised.

Solution

The violated security aspects and offending actions are as follows.

Confidentiality:

E3. Alice revealed her password.

Authenticity:

E1. The attacker spoofed the email.

E2. Alice visited and interacted with the spoofed website specified in the link.

E4. The attacker logged-in to the Web server.

Availability:

E6. The Web server got overloaded.

Integrity:

E5. The attacker invoked many processes on the Web server (*Remark*: a violation of the server's process integrity).

2. Suppose it takes 512 clock cycles to test whether a 64-bit cryptographic key is correct, when given a 64-bit plaintext and its corresponding ciphertext.

   (a) How long does it take to exhaustively check all the keys using a 4GHz (single-core) processor?

   (b) How long does it take on a cluster of 1024 servers, each with a quad-core 4Ghz processor.

   (*Hint*: For simplicity, you can take 1 year $\approx 2^{25}$ seconds. Also note that: 1K $= 2^{10}$, 1M $= 2^{20}$, 1G $= 2^{30}$.)

   Solution for (a)

   Notice that a 4GHz processor has $2^2 \cdot 2^{30} = 2^{32}$ cycles per second.
   From the problem description, testing 1 key takes $512 = 2^9$ cycles.
   In 1 second, the processor can thus check $2^{32} \,/\, 2^9 = 2^{23}$ keys.
   To check all 2^{64} keys, the processor needs $2^{64} \,/\, 2^{23} = 2^{41}$ seconds.
   Since 1 year $\approx 2^{25}$ seconds, the total time needed is therefore: $2^{41} \,/\, 2^{25} \approx 2^{16} \approx 2^6 \cdot 2^{10}$ years $\approx$ 64K years.

   Solution for (b)

   Given 1024 servers, each with a quad-core processor, we thus have $1024 \cdot 4 = 2^{10} \cdot 2^2 = 2^{12}$ processors.
   The total time needed is now reduced by a factor of 2^{12} to become: $\approx 2^{16} \,/\, 2^{12} \approx 2^4 \approx 16$ years.

3. Suppose it takes 512 clock cycles to test whether a 32-bit cryptographic key is correct, when given a 32-bit plaintext and its corresponding ciphertext.

   How long does it take to exhaustively check all the keys using a 4GHz (single-core) processor? Using exhaustive search, is it then possible to crack a ciphertext and obtain its plaintext in realtime?

   Solution

   Notice that a 4GHz processor has $2^2 \cdot 2^{30} = 2^{32}$ cycles per second.
   From the problem description, testing 1 key takes $512 = 2^9$ cycles.
   In 1 second, the processor can thus check $2^{32} \,/\, 2^9 = 2^{23}$ keys.
   To check all 2^{32} keys, the processor needs $2^{32} \,/\, 2^{23} = 2^9 = 512$ seconds ≈ 8.5 minutes.
   The plaintext thus cannot be recovered in realtime.
   (*Note*: Nevertheless, the calculation shows how a 32-bit key can be broken rather easily by the processor).

Now consider a walkie-talkie system called *Secure Walkie Talkie* (SWT)*, which encrypts its communication using a 32-bit symmetric keys k. In each communication session of SWT, the first 64 bits of the plaintext are always the string of zeros, and the last 64 bits the string of ones. Given a plaintext m and the key k, the encryption is done in the following way:

(a) Randomly choose a 32-bit IV;

(b) Compute $\widetilde{k} = IV \oplus k$;

(c) Use a stream cipher to encrypt the plaintext m with $\widetilde{k}$ as the secret key, and output the ciphertext c;

(d) Transmit the IV, followed by the ciphertext c, over the air.

We assume that attackers can eavesdrop and capture all ciphertexts (including the IVs) transmitted over the air. We know that a 32-bit key is too short, and can be broken. However, as calculated above, it would take a relatively long time. In their marketing efforts, SWT thus claims that its 32-bit key is sufficient for many applications. This is what appeared in their advertisement: "The 32-bit key is sufficient. By the time your message is maliciously decrypted, it already becomes useless".

Now, you want to design a hand-held device that is able to crack SWT system and obtain its plaintexts in *realtime*. The hand-held device can have computing resources comparable to a mobile phone. Note that in order to achieve its objective, the device should be able to determine the employed 32-bit secret key readily (say within 0.1 second) when given a ciphertext. Suggest a way to derive the secret key very fast.

(*Hint*: Assume that the hand-held device can hold a large, say 32GB, of pre-computed table whereby the key can be looked up.)

Solution

Utilize the *time-space tradeoff*, i.e. solving the problem in less time by using more storage space, as follows:
Construct a table of $Enc_{\widetilde{k}}(000\ldots000)$ for all possible 2^{32} values of $\widetilde{k}$. Notice that there are 2^{32} entries in the table, where each entry is 64-bit long as the result of encrypting the 64-bit of zeros. Hence, the derived table will take $2^{32} \cdot 64$ bits $= 2^{32} \cdot 8$ bytes $= 2^{32} \cdot 2^3 = 2^2 \cdot 2^{30} \cdot 2^3 = 2^2 \cdot 2^3 \cdot 2^{30} = 2^5 \cdot 2^{30}$ bytes = 32GB.
(Storing the values of $\widetilde{k}$ in the table, if required, additionally takes 16GB.)
To break a SWT communication, first extract the first 64 bits of the captured ciphertext, then perform a lookup operation on the constructed table in order to determine the employed $\widetilde{k}$.
Given $\widetilde{k}$, the attacker can perform a decryption process using the stream cipher, thus recovering the plaintext in realtime.

(Note: Notice that, from the way the stream cipher is used in SWT, knowing $\widetilde{k}$ is sufficient to recover the plaintext.
If desired, the attacker can also recover k from $\widetilde{k} = IV \oplus k$ by calculating: $IV \oplus \widetilde{k} = IV \oplus IV \oplus k = 0 \oplus k = k$.)

4. Lecture 1 mentioned that Winzip can encrypt a compressed file. Why it is meaningless to carry out the two operations in the other way, that is, first encrypts the file, and then compresses the encrypted file?

   (*Hint*: Consider the effectiveness of compression on "random" sequences, and also a requirement of a good encryption scheme.)

Solution

Compressing an encrypted file will yield very little or no compression gain.
This is since the encrypted file will resemble a "random" sequence (due to a requirement of a good encryption scheme).
A compression algorithm, which takes advantage of repeating patterns, therefore will not work well on an encrypted file.

5. Bob encrypted a video file using Winzip, which employs the 256-bit key AES. He chose a 6-digit number as password. Winzip generated the 256-bit AES key from the 6-digit password using a "hash" function, say SHA1.

   Alice obtained the ciphertext. Alice also knew that Bob used a 6-digit password. Given a "guess" of the 256-bit key, Alice could determine whether the key successfully decrypted the file.

How many guesses did Alice really need in order to get the video from the ciphertext encrypted with a 256-bit key in this case?

Solution

Despite the use of the 256-bit AES key, the total number of guesses needed is only $10^6 = 1$ million.
(Note: Why not 2^{256}?)

6. Find out more about these terminologies:
   - *Cryptology, Cryptanalysis, Cryptography,*
   - *NSA, NIST, Cryptography backdoor, Key escrow, Decryption order.*

Solution

Cryptology = **Cryptography** + **Cryptanalysis**.

The National Security Agency (NSA): a national-level intelligence agency of the US Dept of Defense, which is responsible for global monitoring, collection, and processing of information and data for foreign intelligence and counterintelligence purposes, specializing in a discipline known as signals intelligence (SIGINT).

The National Institute of Standards and Technology (NIST): a measurement standards laboratory, and a non-regulatory agency of the US Dept of Commerce, whose mission is to promote innovation and industrial competitiveness.

Cryptography backdoor: a method, often secret, of bypassing normal encryption in a cryptosystem. It allows an intruder to access the plaintext without having the correct user credentials.

Key escrow: an arrangement in which the keys needed to decrypt encrypted data are held in escrow so that, under certain circumstances, an authorized third party may gain access to those keys.

Decryption order: an order that forces suspects to decrypt their encrypted data or give up their keys.

Find out more about the following well-known persons in cryptography:

- *Whitfield Diffie, Ron Rivest, Alice, Bob, Eve, Mallory*, and *Trent.*

Solution

Whitfield Diffie: one of the pioneers of public-key cryptography; co-inventor of Diffie-Hellman key exchange; won the 2015 Turing Award.

Ron Rivest: co-inventor of the RSA algorithm; inventor of the symmetric-key encryption algorithms RC2/RC4/RC5; inventor of the MD2/MD4/MD5/MD6 cryptographic hash functions; co-author of "Introduction to Algorithms" book; won the 2002 Turing Award.

Alice, Bob, Eve, Mallory: Check `https://en.wikipedia.org/wiki/Alice_and_Bob#Cast_of_characters`.
Trent (or **Ted**): a trusted arbitrator as a neutral third party.

(*Optional*) Consider the following questions:

- Can NSA break AES?
- Can NSA by-pass cryptography?

Solution

There are no right or wrong answers to the questions. They are just to stimulate discussions. You can freely speculate on the issues.

*: Companies are purely fictional.

— End of Tutorial —