Challenge Proposal for the Hacking Challenge Creation

Predictable Session IDs & Steganography

Alexander Daniel Nikolaos Lelidis, André Baptista Águas,
Ard Kastrati, Khalid Aldughayem, Uroš Tešić

November 28, 2017

Abstract

This challenge combines two different tasks from Web Security and Steganography. It addresses the problem of the predictable sessions and the security through obscurity. The idea of this challenge is first to bypass the authorization in the HTTP level by hijacking a session with a predictable session ID. Secondly, the attacker must extract the secret information which is hidden by using steganography instead of cryptography.

1 Requirements

- A program to reverse the Mersenne Twister.
- Basic knowlegedge about steganography.

2 Learning goals

There are three different learning goals in this challenge:

1. *If the attacker can control the pseudorandom generator, then he can control everything.* As discussed in the lecture, controlling the pseudorandom generators is the perfect attack, since it is almost impossible to detect. Generating not truly random numbers are very hard to detect.

2. *The security of an algorithm should rely solely on the secrecy of the key* [1]
Never use algorithm whose security relies solely on the fact that the algorithm is secret. One speaks of security by obscurity. Such algorithms are unflexible and insecure from todays point of view. For example, if the algorithm is somehow revealed, then the whole algorithm should be changed. Next, all the parties that partake in the communication must know the algorithm, meaning they can decrypt all the other participants messages. With security through obscurity it is also not possible to establish standard algorithms, which can be tested from others. As a countermeasure, Kerckhoffs's principles have been established. In 1833, Auguste Kerkhoffs stated six design principles for military ciphers in [2] and one of them is [1]:

   - The cipher method should not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience.

   It is worth mentioning that this Kerkhoff's principle has been often ignored, which led to fatal results.

3. *Adding more different unsecure mechanisms in the system doesn't make it more secure, since the attacker can simply bypass all of them one after another.*

3 Mission

Alice and Bob need to share a secret with each other. However, they don't trust the security of the chat in the website "Mission impossible". One day Alice reads about the *steganography* and she was amazed. As a beginner in modern cryptography, she thought that this has to be the best way to share secrets. Without further thinking she arranges a meeting with Bob and they decide to hide their data in different pictures that they send to each other. Alice was right about one thing: *The website is indeed not secure and there is a way to bypass the login page.* Ironically, the *steganography* is not much more secure than that. Hence, adding *steganography* as a measure of security won't help them. Your mission, should you choose to accept it, is to bypass the login page for *Alice* or *Bob*, read the data exchanged between them and extract the secret message that they exchanged with each other.

4 Mitigation

1. Use cryptographically secure pseudorandom number generator (CSPNR).
2. Use standard encryption schemes from modern cryptography (instead of steganography) for confidentiality (such as AES, RSA, El-Gammal etc).

5 Type of Challenge

Online

6 Category

Web Security & Cryptography

7 Hints

- Mersenne Twister is used to generate the session keys!
- The secret data is hidden in only some of the bits in every pixel.

8 Step-by-step instructions

- Generate 624 session IDs by logging in 624 times.
- Get the state of the generator of the Mersenne Twister
- Predict the next session ID.
- Wait for *Alice* to log in[1].
- Log in by using the session ID (Session hijacking)
- User logs into website where he sees the chat and images
- Extract the location by using the LSB of each pixel.

[1] In order for this to be possible, we plan to implement that every 15 minutes *Alice* logs in the website.

REMARK. Another way to predict session IDs is to provide a feature in the application level that issues random numbers. We instantiate for some basic feature in the application level a **different** pseudorandom generator but with the **same seed** that was used for the generation of the session IDs (e.g. we instantiate for each chat the same pseudorandom generator that was used for the session ID generation and for every message that is sent within the chat a new random number number is generated). This way, the attacker can generate all random numbers (which are used for session IDs) by only sending arbitrary messages in one arbitrary chat. He can then know all session IDs and find the one that belongs to *Alice* by simply trying them all. In case the reversing process of the Mersenne Twister is a tedious work for the attacker we can change it to this alternative.

References

[1] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography (Chapman & Hall/Crc Cryptography and Network Security Series)*. Chapman & Hall/CRC, 2007.

[2] Auguste Kerckhoffs. La cryptographie militaire. *Journal des sciences militaires*, IX:5–83, January 1883.