Enhanced Password Security: PAKE and Anti-Phishing

Sunday 29th October, 2023 - 16:31

Meireles Lopes Steve
University of Luxembourg
Email: steve.meireles.001@student.uni.lu

This report has been produced under the supervision of:
Skrobot Marjan
University of Luxembourg
Email: marjan.skrobot@uni.lu

Abstract—In today's world, password security is very important. Most people have weak passwords which exposes them to brute force attacks. Furthermore malicious attackers also often use phishing attacks. These attacks expose a huge vulnerability and therefore a risk for the data of a lot of people. This project is to answer the scientific question: In the context of client-to-server authentication over an insecure network, how does one detect if a password file on the server side (containing usernames and passwords) was compromised by intruders and at the same time prevent phishing attacks on clients?

1. Main required competencies

1.1. Scientific main required competencies

For the scientific deliverable, it is essential for the student to possess a solid foundation in cryptography and security. This includes a thorough understanding of key concepts such as password security, symmetric and asymmetric encryption, and key exchange protocols. The student should be well-versed in the mechanisms that underpin these concepts, enabling them to analyze, design, and implement secure authentication systems.

In addition to cryptographic knowledge, a grasp of mathematical concepts is imperative. Competency in areas such as set theory and mathematical notations is essential to navigate the mathematical underpinnings of encryption algorithms and security protocols. These mathematical skills are integral to the development and analysis of secure authentication mechanisms.

1.2. Technical main required competencies

The technical deliverable, while sharing the core competencies with the scientific counterpart, demands an additional skill set. In addition to cryptography and security expertise, the student tasked with the technical deliverable must be proficient in the programming language Python.

Python is the chosen medium through which the theoretical concepts will be translated into practical, functional solutions. The student should have a strong command of Python programming, enabling them to effectively implement authentication systems, manage password files, generate decoy passwords, and integrate the SweetPAKE protocol.

These technical competencies encompass not only an understanding of cryptographic principles but also the ability to translate this knowledge into tangible, working software. Proficiency in Python is the foundation that bridges theoretical security concepts to real-world application, making the deliverable a practical and impactful contribution to the field of authentication security.

2. A Scientific Deliverable 1

In the realm of cybersecurity, a significant number of enterprises rely on username and password logins as the primary means of authenticating their users or employees. This method extends a wide array of technologies and systems including from email services to online banking. Everybody has a few technologies that require them to authenticate with their username and password. However is commonly known that this is often exploited by attackers through weak passwords, brute forcing or even phishing attacks. It has a lot of vulnerabilities for example weak passwords, often a result of human tendencies to choose easily guessable combinations, present a security risk. The threat is not limited to weak passwords but also by various tactical attacks including brute force attacks, where automated scripts or tools systematically guess passwords until they find the correct one. Also popular are phishing attacks, which involve social engineering techniques, where criminals disguise themselves as trustworthy parties to deceive individuals into revealing details about them or their login credentials. This can take the forms of fake websites, fake emails or misleading messages.

2.1. The scientific question

These vulnerabilities are particularly pronounced in the context of client-to-server authentication over insecure networks, where the transmission of login credentials is potentially exposed to eavesdropping and malicious interception. This brings us to the scientific question: In the context of client-to-server authentication over an insecure network, how does one detect if a password file on the server side (containing usernames and passwords) was compromised by intruders and at the same time prevent phishing attacks on clients? The deliverable embarks comprehensive exploration of various security approaches. The project encompasses three key areas: password security, and the prevention of brute force and phishing attacks.

2.2. Honeywords and Decoy Passwords

The project will study password security, secure storage of passwords and brute forcing attacks. This will be done by exploring the concepts of "honeywords" as introduced in the Honeyword paper [3] by Juels and Rivest. The paper introduces how you can add decoy passwords to the password file and suggests methods to generate these passwords such that even a skilled attacker can not guess the correct password from the decoy passwords. Then it introduces the concept of a honeychecker which stores the index of the correct passwords and communicates with the server that's working with the client.

2.3. PAKE

Furthermore, it looks into ways to enhance current password authentication protocols from sending a hash of a user's password over TLS to utilising PAKE (Password Authentication Key Exchange). This transition to PAKE introduces a a robust layer of security that prevents phishing attacks by ensuring a secure client-to-server communication.

2.4. SweetPAKE protocol

On top of all that the paper is going to examine the SweetPAKE protocol from Arriage, Ryan and Skrobot [1] which uses PAKE and enhances it with a decoy password which adds a password file leakage detection mechanism at the server side. The SweetPAKE uses PAPKE[2] (Password-Authenticated Public Key Encryption) principles, these principles are going to be analyzed and described.

2.5. Deliverable Structure

To answer the question, the deliverable will be split into three main sections. The first section will try to answer the question: How to detect if a password file is compromised? This will cover in detail how decoy passwords work and how an external server can alarm the administrator if an attacker compromises the password file. The section also talks about the different decoy password generation methods and their probabilities of being guessed.

The second section goes into detail about the technologies of PAKE. How it works and which protocols are being used nowadays. This will be done with graphs showing visually the process and the section will explain the mathematical side of such protocols. PAKE provides several properties, which are that a man-in-the-middle does not have enough information to execute a brute force guessing attack. Which therefore provides strong security using weak passwords.

The last section will explain how to improve the security of the methods talked about in the first section by adding a PAKE protocol in the client-server authentication. It goes into detail about the SweetPAKE [1] protocol and PAPKE[2]. This makes an attack a lot harder and it provides additional protection against phishing attacks to the already strong idea established by Juels and Rivest. The section may also have a benchmark of the implementation done in the technical part of this paper.

2.6. Conclusion

In a landscape defined by the reliance on username and password-based authentication and the security of these credentials is very important. This project's exploration into honeywords, PAKE protocols, and the SweetPAKE enhancement offers a promising avenue to strengthen password security and prevent phishing attacks. By combining innovative strategies, we aspire to elevate the standards of authentication and significantly enhance the security of digital interactions.

3. A Technical Deliverable 1

The technical deliverable will be the implementation of a decoy password generator according to proposed protocols from the Honeyword paper [3] of Jules and Rivest which is discussed in the scientific part. Additionally, it integrates the SweetPAKE protocol [1] suggested by Arriaga, Ryan and Skrobot.

3.1. The programming language

The entire technical part is realized through the versatile and widely used programming language Python. Python offers an ideal platform for these implementations due to its ease of use, extensive libraries, and cross-platform compatibility. It ensures that the solution remains accessible to a wide range of users and organizations.

3.2. Motivation and Significance

This project is beneficial to the security world considering there are not a a lot of open-source PAKE implementations despite its security value. Combining PAKE with a decoy password generator which further enhances password security does not exist in the current market. It has the potential to reshape the landscape of digital security. It not only strengthens traditional password security but also prevents phishing attacks by malicious parties. It is a formidable asset in the ongoing battle against malicious attacks in our interconnected world.

3.3. Project structure

The deliverable will consist of two parts and try to combine both. The first part is the implementation of the ideas of Juels and Rivest [3]. The second part is the implementation of the SweetPAKE introduced by Arriage, Ryan and Skrobot [1].

3.3.1. Part 1: Honeywords Implementation. The aim of the project is to give a practical example of how the suggested password-generation methods of Juels and Rivest can be implemented. In addition to that it will simulate the actions of the Honeychecker (extern server).

The functions the first part has to have:

Firstly, there will be a password file and the deliverable has to be able to add a user to it. The function add_user(name, pw) will be responsible for that. This function takes a name and a password as a parameter. It generates an array of honeywords consisting of real passwords and decoy passwords. After generating, the program will shuffle the array with the Fisher-Yates algorithm. Then it writes the username and the passwords separated by a ":" into the password file.

Secondly, the Honeychecker functions. The Honeychecker checks(i) if the integer i corresponds to the index of the correct passwords. It can also receive(i) a new index when a user changes passwords or a new user is added to the database. This function changes or adds the new index of the user to the index file (the file which stores the indexes of the correct password of the corresponding users).

The function of the deliverable is coded as suggested by the honeyword paper such that it can be said to be secure.

3.3.2. Part 2: SweetPAKE implementation. The second part will consist of a PAKE implementation. This will ensure security against phishing attacks. Two parties can safely exchange long-term keys without a middleman being able to trick one of the parties.

The functions of the second part:

Two parties have to agree on a key. The project will be able to communicate and agree on a session key this will be done following the SweetPAKE paper [1]. This involves several steps and functions with the mathematical background behind it.

After successfully implementing a PAKE, the SweetPAKE will be fully implemented by combining both previous parts. This implementation will then be a good example of an implementation of both Honeywords and SweetPAKE.

3.4. Bonus

In case the student has time, a benchmark may be added to the SweetPAKE protocol which tests the speeds of several different encryption algorithms and the difference between the speed of using the SweetPAKE protocol and using the current methods.

3.5. Existing code used

To be able to achieve this program will use the open-source project python-spake2 of Warner as a reference. It will use the standard library random and hashlib to encrypt and generate passwords.

3.6. Conclusion

In conclusion, this technical deliverable merges the power of Honeywords and the SweetPAKE protocol to create a robust and innovative approach to authentication security. It offers a practical solution to enhance password security, prevent phishing attacks and strengthen the authentication process in an interconnected digital world.

References

[1] Ryan Arriage and Skrobot. "SweetPAKE: Key exchange with decoy passwords". 2023.

[2] Tatiana Bradley et al. "Password-authenticated public-key encryption". In: Applied Cryptography and Network Security: 17th International Conference, ACNS 2019, Bogota, Colombia, June 5–7, 2019, Proceedings 17. Springer. 2019, pp. 442–462.

[3] Ari Juels and Ronald L Rivest. "Honeywords: Making password-cracking detectable". In: Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security. 2013, pp. 145–160.