|  |
| --- |
|  |
| Capstone Experience IST 894  Carl Laneave |
| Lab 8 Report |

Table of Contents

[1.0 Introduction – Cryptography 3](#_Toc148908856)

[1.1 Lab Results – Cryptography 4](#_Toc148908857)

[2.0 Introduction – AAA 9](#_Toc148908858)

[2.1 Lab Results – AAA 10](#_Toc148908859)

[4.0 References: 16](#_Toc148908860)

[5.0 Activity Log 17](#_Toc148908861)

# Introduction – Cryptography

During the execution of this lab, an evaluation was done of Steganography as well as Cryptography. Throughout the lab, different types of Cryptography were tested to show how files and images can host secrets that required different types of decryptions to access. The most common tool used for encryption as well as decryption was done through OpenSSL. OpenSSL allows for the creation of many different types of Cryptography. This includes DESC ECB, DES CBC, AES CBC, AES ECB, and more. These different types of ciphertext encryption methods are critical in the protection of sensitive data in a file.

There was also an evaluation of image metadata using Steghide, a tool popular for its ability to do steganography on hidden data inside of images. Hidden data inside images can be found when using Steghide CLI as well as hide data into the images themselves. Not only does Steghide allow for hiding information in an image, but it also allows for doing it in audio files as well.

Lastly, it's important to review that there are many different cryptographic algorithms that are used through OpenSSL Ciphers. These ciphers are all important algorithms to understand as any time data needs to be protected, these different types of cryptographic algorithms can provide different types of encryptions. Through understanding how each algorithm works, a security engineer can get a better understanding of how to encrypt a file, which algorithm to select, and how to access that said data for future usage.

# 1.1 Lab Results – Cryptography

A screenshot of a computer

Description automatically generated

Figure 1.0 – Open jpg image using EOG CLI.

A computer screen shot of blue text

Description automatically generated

Figure 1.1 – Extract data from image using steghide.

A screenshot of a computer screen

Description automatically generated

Figure 1.2 – Run command to get ciphertext of each file in the folders.

A screen shot of a computer

Description automatically generated

Figure 1.3 – Use OpenSSL to decrypt the DES ECB Ciphertext.

A screen shot of a computer

Description automatically generated

Figure 1.4 – Use OpenSSL to decrypt the DES CBC Ciphertext.

A computer screen with white text

Description automatically generated

Figure 1.5 – Use OpenSSL to decrypt the 3DES ECB Ciphertext.

A computer screen with white text

Description automatically generated

Figure 1.6 - Use OpenSSL to decrypt the 3DES CBC Ciphertext.

A screen shot of a computer

Description automatically generated

Figure 1.7 - Use OpenSSL to decrypt the AES ECB Ciphertext.

A computer screen with white text

Description automatically generated

Figure 1.8 - Use OpenSSL to decrypt the AES CBC Ciphertext.

A computer screen shot of a computer screen

Description automatically generated

Figure 1.9 - Create an RSA Key pair using OpenSSL key generation.

A screen shot of a computer

Description automatically generated

Figure 1.10 – Use the created RSA Key to encrypt and decrypt a txt file

A screenshot of a computer program

Description automatically generated

Figure 1.11 – Use OpenSSL to make a new key with passphrase secure.

A screenshot of a computer

Description automatically generated

Figure 1.12 – Create an ECC keypair with OpenSSL using Secp256k1 as the elliptic curve parameter.

A screen shot of a computer

Description automatically generated

Figure 1.13 – Create ECC public and private keys using OpenSSL.

A computer screen shot of a black screen

Description automatically generated

Figure 1.14 – Create an ECC Digital Certificate using OpenSSL.

A black screen with a check mark

Description automatically generated

Figure 1.15 – Lab is complete.

# 2.0 Introduction – AAA

In this lab, we evaluate the usage of SSH key generation and how SSH keys are used to access hosts and other servers. SSH keys are a fundamental approach in which users can connect to different hosts/servers through the usage of public and private pem keys. These keys allow users to not only protect their connections but also protect the server by only allowing users with approved submitted keys on the client and host to connect.

The next piece was a focus on password management using KeePass. Password management is a critical tool that allows us to store sensitive information such as passphrases, keys, and more in a hosted database. This database is then only accessible by the user who submitted the original secrets. All secrets are protected by passcodes and/or keys. Many other tools such as Hashicorp Vault have extended these types of credentials into a more complex system that allows users to get ‘Just in Time’ credentials that are ever rotating to prevent leaking sensitive information.

Lastly, we looked at the importance of proper Linux file permissions. File permissions in Linux are a critical path in protecting sensitive files and folder constructs. Files and folders can be shared between users and groups, as well as restricted to only certain users, or even completely blocked off for only root access. Permissions can include read, write, modification, and deletion rights which are all completed using the chmod command. Users can also see file permissions by using command line ls flags to list the overall permissions. This is a critical skill set that any security engineer, as well as system administrator, to master as it can be one of the most important areas of protection and security on a server.

# 2.1 Lab Results – AAA

A screenshot of a computer

Description automatically generated

Figure 2.0 – Generation a SSH key and start the SSH service.

A computer screen shot of a black screen

Description automatically generated

Figure 2.1 – Generate an SSH key on the student host computer.

A screenshot of a computer program

Description automatically generated

Figure 2.2 – SSH into the server and copy the root key. Once done, log in using the root key and verify success as root.

A screenshot of a computer

Description automatically generated

Figure 2.3 – Leave the SSH connection and open KeePass.

A screenshot of a computer

Description automatically generated

Figure 2.4 – Look at current directories in KeePass and create an SSH directory.

A screen shot of a computer

Description automatically generated

Figure 2.5 – Add new entry into SSH directory.

A screen shot of a computer

Description automatically generated

Figure 2.6 – Exit the editor and add the entry to the KeePass Database.

A screenshot of a computer

Description automatically generated

Figure 2.7 – Load SSH folder contents and show stored secrets.

A screenshot of a computer

Description automatically generated

Figure 2.8 – Load credentials from KeePass Database and display.

A screenshot of a computer screen

Description automatically generated

Figure 2.9 – Load the Mandatory access control as root in the iptables.

A black screen with white text

Description automatically generated

Figure 2.10 – Add in a new input for CIDR Range 172.20.0.0/16 to the iptables.

A screen shot of a computer program

Description automatically generated

Figure 2.11 – Check inspections are working properly and see permissions on the file.

A computer screen shot of a computer program

Description automatically generated

Figure 2.12 – Login as admin and check permissions on a file for guest user. Access is denied.

A computer screen shot of a black screen

Description automatically generated

Figure 2.13 – Change the permissions to allow read for admin.

A screen shot of a computer

Description automatically generated

Figure 2.14 – Switch to guest user, verify they can see developer notes text file.

A screen shot of a computer program

Description automatically generated

Figure 2.15 – Change permissions on the text file as the developer revokes any read rights outside them.

A black screen with a check mark in a circle

Description automatically generated

Figure 2.16 – Lab is complete.

# 4.0 References:

[1] *KeePass*. (2022, July 13). Wikipedia. <https://en.wikipedia.org/wiki/KeePass>

‌[2] *Permissions in Linux* (2017) *GeeksforGeeks*. Available at: <https://www.geeksforgeeks.org/permissions-in-linux/>.

[3] Bibi, K. (n.d.). *Steghide tutorial for beginners*. <https://linuxhint.com/steghide-beginners-tutorial/>

[4] *Difference between Steganography and Cryptography*. (2019, May 21). GeeksforGeeks. https://www.geeksforgeeks.org/difference-between-steganography-and-cryptography/

‌

‌

‌

‌

# 5.0 Activity Log

| **Member Name** | **Task Date** | **Task Details** |
| --- | --- | --- |
| Carl Laneave | 10/22/2023 | Created Template, executed all labs, took screenshots, and completed report |
|  |  |  |
|  |  |  |