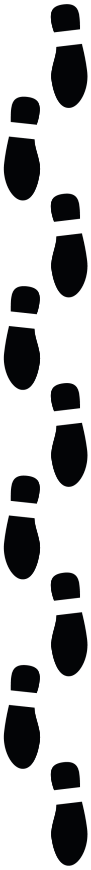
HackTrick 2024 Documentation

Dell Technologies

March 2024



Contents

List of Figures

1 Introduction

Hello, and welcome to Hacktrick 2024!

Congratulations on successfully completing all the previous phases and finally making it to the game - the most interesting part! We are delighted to have you all here.

This document will provide you with all the necessary information from start to end to participate in this year's Hacktrick. For clarity and ease of use, there are two additional supporting documents that focus on specific areas. They will be referenced where relevant.

Moving forward, the hackathon will be divided into two phases, and the details of each phase will be mentioned at the end of this document. Both the student and professional tracks will run concurrently, with contestants from each track being separated for fairness.

Without further ado, let's get started!

1.1 Game Description

This year, the Hackathon revolves around Steganography.

steganography
/stɛɡəˈnɔɡrəfi/
noun
the practice of concealing messages or information within other non-secret text or data.
An image, in our case.

Contestants get to compete in teams against each other, taking one of two roles at a time but ultimately playing on both sides. One trying to send a secret message, and the other trying hard to intercept it.

Firstly: The Fox. Mischievous, and sly, the Fox uses all its tactics to fool the Eagle and try to send the message through to the parrot using steganography. As the Fox, you'll have the opportunity to invest your time wisely into honing your skills and creating distractions to increase your chances of evading the Eagle's watchful gaze.

Second: The Eagle. Sharp-eyed and vigilant, the Eagle uses its attentiveness to try to intercept and decode the messages sent without getting fooled. Beware of the Fox's devious tricks, for Fake messages may cross your path. Your mission is to distinguish truth from deception, ensuring that only genuine messages are intercepted while avoiding costly mistakes.

The parrot represents the game administrator that receives the messages and scores both ends accordingly.

Read on for further VERY INTERESTING details!

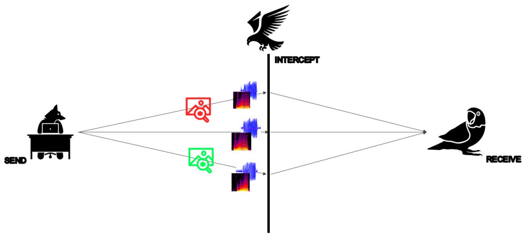


Figure 1: Game overview

1.2 Game Environment

By now, every team has received their team ID via email.
The team ID is crucial as it will serve as the primary identifier throughout the game. You will interact with the server using the assigned IP address and utilize the provided APIs, all of which can be found in the accompanying API documentation. Aside from that, this documentation will be your go-to resource for understanding everything about the game and strategies for winning.
Here is a diagram to give you an overview of the game, we will be discussing every part in detail following that.

2 Fox

Let's start with the Fox, our primary player.
The primary objective for the Fox is to send the secret message to the parrot, encoded through steganography, while devising a strategic game plan to outsmart the Eagle and prevent it from intercepting the message. For each game, you will be provided with a message of a specific length and an image to use for encoding the messages. During the first phase, the message length is fixed to 20 characters.

2.1 Overview

The game is played in chunks (anything between 1 and 20). In every chunk, there are 3 channels that concurrently carry your messages. The messages can be one of the following 3 types:

- Real: These messages are part of the original message intended for transmission.
- Empty: These are Empty messages.
- Fake: These are Fake messages used to deceive the Eagle. Unlocking this feature requires solving riddles - details of which are mentioned in section 2.5.

2.2 Chunks

As the Fox, you have the freedom to divide the message into one or more chunks. However, it's important to note that there is a time limit for completing your part of the game. Failing to send the full message within the allocated time will result in a score penalty. It is very important to make sure you send the message exactly as it is. The order of the messages as well as the characters sent should precisely match the given message.

Rules

- You can only send a maximum of one Real message per chunk.
- The maximum number of chunks you can use is 20.
- You can only use Fake messages if your budget allows for it (details will be explained in the upcoming sections).
- The rest of the channels would contain Empty messages.
- The messages must be encoded using the Least Significant Bit (LSB) technique, which will be discussed in the next section.

2.3 Steganography: LSB

One common method of steganography involves utilizing the least significant bit (LSB) of digital images. The LSB is the least significant bit of a binary number, meaning it is the bit that has the least impact on the overall value of the number. In a digital image, each pixel is represented by a series of bits, and the LSB is the rightmost bit in that series. By altering the LSB of each pixel or sample, one can embed secret data within the file without changing the overall appearance.

In this approach, the LSB of each pixel in an image is altered to embed hidden data. Since the LSB holds the least amount of information and changes to it are less likely to be noticed by the human eye, this method allows for discrete communication. By replacing the LSB with secret data bits, one can conceal messages within seemingly innocuous files, making steganography a powerful tool for covert communication and data security.

To embed secret data using the LSB method, one must first convert the secret data into a binary format. The secret data is then divided into blocks, and each block is converted into a series of bits. The LSB of each pixel or sample in the image is then altered to match the corresponding bit in the secret data block. This process is repeated for each block of secret data, and the resulting file is then saved.

To extract the secret data from the file, the LSB of each pixel or sample is read, and the corresponding bit in the secret data block is extracted. This process is repeated for each block of secret data, and the resulting binary data is then converted back into the original secret data.

You are given the implementation of the LSB algorithm and you don't need to implement it from scratch. You are going to use it in:

- The Fox, where you should be able to encode the message in the given image using the least significant bit method.
- The Eagle, where you should be able to decode the message in the given image using the least significant bit method.

2.4 Fake messages

2.4.1 Main Idea

When you, as the Fox, decide to send a message through any of the 3 channels, you have the option to send a Real, Fake, or Empty message. The purpose of sending Fake messages is to impede the Eagle's progress. If the Eagle falls for a Fake message and decides to intercept it, their score will be negatively affected, hindering their collection of the secret message being transmitted. However, it's important to note that Fake messages are not readily available to you. In order to earn them, you must solve riddles that are provided.

2.4.2 How to Increase Your Budget

To increase your budget of Fake messages, you can choose to solve any of the provided riddles. For every correct riddle solution submission, you will be awarded a predefined number of Fake messages. The reward you receive depends on the riddle solved and is contingent upon the accuracy of your solution. Once you have been granted Fake messages, you can strategically use them among the Real messages to attempt to deceive the Eagle.

2.4.3 Limitations on Use

Although you can solve as many riddles as you want to increase your budget, you can only use a **maximum** of **12** Fake messages throughout the entire game. Please note that the Fake message budget is valid only for a single game and is reset to zero at the beginning of the next game, where you will have to re-solve riddles if desired. If you have any unused budget at the end of the game, you will be rewarded for it.

2.4.4 Utilizing the Fake Messages

When sending messages, you need to specify several elements, including the **"message_entity"**, which represents the character **'F'**, **'R'**, or **'E'** depending on whether the message sent is Fake, Real, or Empty. Selecting **'F'** indicates that the message is a Fake one, and decrements your Fake messages' budget. There is no restriction on the number of Fake messages you can send per chunk, as long as your remaining budget covers it. You can send anywhere between **0** and **3** Fake messages, depending on your chosen strategy.

However, if you try to send a Fake message when your budget is 0, the server will automatically replace it with an Empty message before sending it to the parrot. Additionally, it's crucial to note that misusing this feature and falsely claiming a Real message as Fake, or vice versa, will result in **severe** penalties. Whether it's an intentional attempt to deceive the Eagle or a genuine mistake, any unlawful attempts to deceive the other player are against the rules and will be subject to significant penalties.

That will result in the following sequence diagram for requesting the riddles:

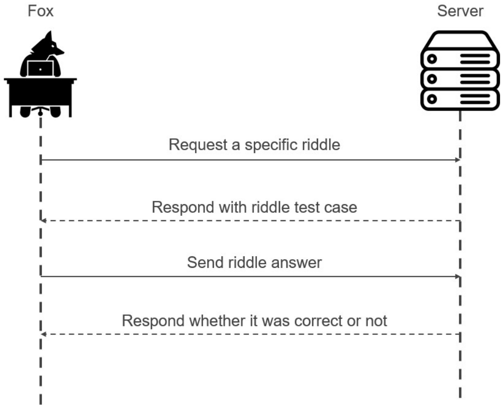

Fox
Server
Request a specific riddle
Respond with riddle test case
Send riddle answer
Respond whether it was correct or not

Figure 2: Fox riddle request sequence diagram.

2.5 Riddles

2.5.1 Main Idea

As mentioned in the previous section, the riddles exist as a side quest for you to solve to gain a budget of Fake messages. You can choose to solve riddles from the available pool in the Riddles Documentation (external document). The reward received depends on the chosen riddle and the correct submission of the solution within the allocated time specific for this riddle. When you submit a riddle solution, the server will provide an update on your status, including your Fake message budget.

2.5.2 Riddle Rewards

Whenever you successfully solve a riddle, you will receive a specific reward based on the difficulty level of the riddle. The rewards for each difficulty are as follows:

- **Easy:** a budget of **1** Fake message.
- **Medium:** a budget of **2** Fake messages.
- **Hard:** a budget of **3** Fake messages.

There is a total of **10** different riddles:

- **2 Security Riddles** (medium and hard).
- **3 Computer Vision Riddles** (easy, medium, and hard difficulties).
- **2 Machine Learning Riddles** (easy and medium difficulties).
- **3 Problem Solving Riddles** (easy, medium, and hard difficulties).

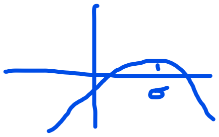
In the case of solving all riddles correctly, the maximum number of Fake messages you can obtain is **20**. However, to ensure fairness among the teams, the maximum number of Fake messages allowed to be used in a single game is capped at **12**. This ensures that teams are not expected to solve all riddles in all domains. Any unused Fake message budget will still be rewarded as previously mentioned, and as indicated in the scoring function.

Remember, riddles are only optional side quests, but they can enhance your game strategy. The details of each riddle can be found in the provided **Riddles Documentation**.

Best to use 6 msg fake

2.6 Scoring Function

Here is the scoring function for the Fox:



$$Score = \text{Done}\left(\alpha_1 \frac{\sum\left(\frac{\text{msg}_{\text{Real}}+\text{msg}_{\text{Fake}}}{\text{msg}_{\text{Real}}+\text{msg}_{\text{Fake}}+\text{msg}_{\text{Empty}}}\right)}{n_{\text{chunk}}} + \alpha_2\left(1-\left(\frac{\sum \text{msg}_{\text{Fake}}}{6}-1\right)^2\right) + \alpha_3\left(1-\frac{\text{timeTaken}}{\text{timeOut}}\right) + \alpha_4 \frac{\text{budget}_{\text{remaining}}}{20}\right)$$

The weights are as follows:

$$\alpha_1 : 0.4, \quad \alpha_2 : 0.3, \quad \alpha_3 : 0.2, \quad \alpha_4 : 0.1$$

Done: Represents the successful completion of the message transmission, indicated by a continuous value between 0 and 1.
1st Term: Measures the efficient utilization of the channels based on the number of non-Empty messages sent.
2nd Term: Represents a concave function that evaluates the use of Fake messages in relation to the message size.
3rd Term: Measures the time efficiency by comparing the time taken by the player with the maximum allowed time for the game.
4th Term:Accounts for the remaining budget of Fake messages after the game ends.

Where:

msg$_{\text{Real}}$: The number of Real messages sent.

msg$_{\text{Fake}}$: The number of Fake messages used, limited to a maximum of 12.

msg$_{\text{Empty}}$: The number of Empty messages sent.

n$_{\text{chunk}}$: The number of message chunks sent.

timeTaken: The time taken by the player in this game.

timeOut: The maximum time allowed for the game.

budget$_{\text{remaining}}$: The remaining budget of unused Fake messages at the end of the game.

3 Eagle

3.1 Overview

The Eagle has various responsibilities during the game, and your main objective is to successfully intercept Real messages in order to decipher the hidden message. You have a designated timeout to complete your part of the game, and your efficiency within that time frame contributes to the final score as can be seen in the scoring function at the end of this section.

3.2 Footprints

3.2.1 Main Idea

As previously mentioned, there are **three channels** through which messages are sent **simultaneously** at a time. The channels receive messages in the same order sent by the Fox. However, you can only **intercept one channel** at a time, which means you will **miss** the messages sent on the other **two channels**.

But don't worry, only a **maximum** of **one** genuine message is sent at any given time. So if you can correctly identify the channel on which the genuine message is transmitted, if one exists, you are guaranteed not to miss any Real messages.

To assist in this identification process, you receive **three** footprints at a time, with each footprint corresponding to one of the three channels. These footprints indicate whether the message on a specific channel is Real, Fake, or Empty. By analyzing the footprints, you can **determine** which message is genuine and **request** it from that channel. However, be cautious, as requesting a message that is Empty or Fake would result in a penalty and could lead to missing a Real message altogether!

3.2.2 What it is

Each footprint is a visual representation of an audio file, one for each channel. Your task is to identify the word represented, being one of: **"Dell"**, **Fooled**, or just an **Empty audio** with some **random noise**. Rather than working directly with raw audio files, we've transformed them into **spectrograms** with fixed dimensions **(Tx, Nfreq)**.

The spectrogram can be visualized as a **2D array** that depicts the intensity of the frequency components in the audio file at each moment in time. In simpler terms, you can perceive the spectrogram as a sequence of snapshots, where each snapshot captures the intensity of the frequency components at a particular instant. A frequency component is a singular element that makes up the sound, akin to the vocal cords in human speech.

As mentioned, each footprint is represented as a two-dimensional array (representing the spectrogram). The dimensions of this array are defined by **Tx** and **Nfreq**. In this context, **Tx** stands for the number of time samples, and **Nfreq** represents the number of frequency components in the audio file. For instance, based on the case in figure 3, **Tx = 1998** and

Nfreq = 101.

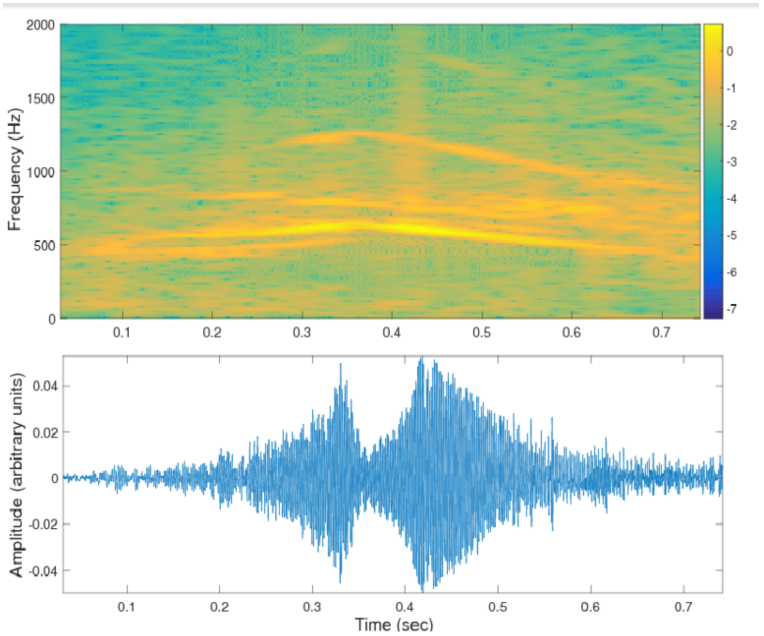


Figure 3: Audio sample with its corresponding spectrogram.

Feel free to use any method you prefer for this task, but we do encourage the use of an AI-driven approach if possible. To train your model, you will be given **2 datasets**: one with **750** samples of Real message footprints, and the other containing **750** samples of Fake message footprints. Your task is to process the given two-dimensional arrays and find the Real ones. You then request the message on that channel, decode it, and send it back to the server. Your decisions will be carefully evaluated and factored into the calculation of the final score. You can see the exact formula later in this section.

3.2.3 Effect on Game

It's important to note that the Fox can potentially mislead you during the footprint detection process. The Fox may strategically invest the budget earned from solving riddles to make it more challenging for you to detect the footprints. Fake messages will add significant difficulty in the footprint detection, and if you incorrectly classify a footprint, you will incur a penalty, providing the Fox with a strategic advantage.

3.3 Requesting The Message

Once you have identified the footprint of a Real message, you should request the message sent on that specific channel by calling the **request message** function and specifying the **channel number** associated with the Real message. If, you believe that none of the channels contains a Real message, you should call the **skip message** function instead. It

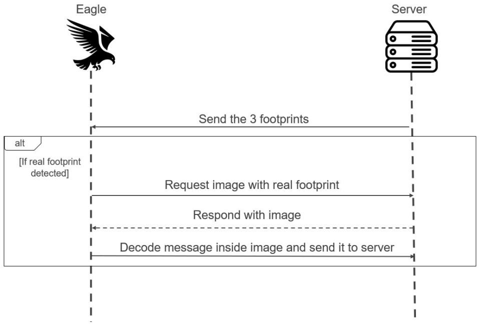

Eagle
Server
Send the 3 footprints
alt
[If real footprint detected]
Request image with real footprint
Respond with image
Decode message inside image and send it to server

Figure 4: Eagle sequence diagram

is important to note that requesting a Fake or Empty message will result in a penalty. The exact scoring function is provided at the end of this section.

3.4 Decoding The Message

Once you have requested and received a message, you need to decode it using the **Least Significant Bit (LSB)** method explained earlier and submit the decoded message. The accuracy of the submitted message will be verified and contribute to your score. It is crucial to remember that after requesting a message, you must submit a message in response. Failure to do so will cause the game to enter a frozen state until the timeout is reached, and the game ends.

Upon submitting or skipping a message, the next set of footprints will be automatically sent to you. If there are no more footprints available - indicating the end of the message - you will be notified, and you should conclude the game. Detailed information about the specific APIs can be found in the attached **API Documentation**.

That pipeline would result in the following diagram: This sequence diagram is for one time step of the game, and that loop ends when you are notified that there are no more footprints.

3.5 Scoring Function

Here is the scoring function for the Fox:

$$Score = (\alpha_1 \text{JaccardDistance} + \alpha_2 \left(1 - \frac{timeTaken}{timeOut + 4t}\right))$$

$$Bonus = Score * 0.2 * \frac{\text{Fake}_{\text{dodged}}}{\text{Fake}_{\text{chunks}}}$$

$$Penalty = Score * 0.2 * \frac{\text{Real}_{\text{missed}}}{\text{Real}_{\text{chunks}}}$$

The weights are as follows:

$$\alpha_1 : 0.7, \quad \alpha_2 : 0.3$$

1st Term: Jaccard distance measures the similarity between your requested messages decoded and concatenated, and the original message sent.
2nd Term: Measures the time efficiency by comparing the time taken by the player with the maximum allowed time for the game.

Where:

$\text{Fake}_{\text{dodged}}$: The number of Fake messages dodged.

$\text{Fake}_{\text{chunks}}$: The number of chunks that included at least 1 Fake message on one of the channels.

$\text{Real}_{\text{missed}}$: The number of Real messages missed.

$\text{Real}_{\text{chunks}}$: The number of Real messages sent.

timeTaken: The time taken by the player in this game.

timeOut: The maximum time allowed for the game.

4 Phases of the Game

4.1 Phase 1: Leaderboard Phase

4.1.1 Details

The first phase, starting today as you receive this document, involves teams playing **15** games as a **Fox**. Scores will be assigned based on their actions, following the previously mentioned scoring function. Teams are allowed to refine their solutions between games, but they will need to solve any desired riddles again since the Fake message budget will reset to **0** after every trial.

Afterward, each team will play **15** games as an **Eagle** against a computerized Fox agent. The score will be updated after each trial, considering the latest and highest score achieved.

$$TotalScore = (0.4 * \text{Fox}_{\text{score}}) + (0.6 * \text{Eagle}_{\text{score}})$$

The team's total score at any point is calculated as a weighted summation of their highest scores in the fox and eagle player modes.

2 leaderboards exist, one for each category (students, and professionals). Every time a team gets a new highscore the respective leaderboard is updated so teams can track their position amongst other teams. The IP address to access the leaderboards can be found in the API documentation.

The **deadline** for submitting all solutions is **11:59 PM on Tuesday, the 5th of March.**

4.1.2 Evaluation

Based on the combined highest scores achieved as the Fox and Eagle on the final day, the top **8** teams in each track will qualify for the next phase.
Please note that after the leaderboard phase ends, you will submit your codes and we will

run it against a plagiarism checker, any team who cheated will be disqualified immediately, and then the announcement of the qualified teams will take place on **Wednesday, the 6th of March**, early in the morning.

4.2 Phase 2: Knockout Phase

After the qualified teams are announced, they will have **Wednesday** and **Thursday** to refine and finalize their solutions, as well as prepare a **business pitch**.

4.2.1 Final Game

On **Friday, the 8th of March**, each team will be randomly matched to compete against another team from the same track. They will play one game as the Fox and another as the Eagle.

4.2.2 Business Pitch

In the closing event, the top **4** teams from each track in this phase will have the opportunity to present their solutions to Dell Technologies' leaders. Out of these teams, **3** will be selected as the hackathon winners.
The pitch accounts for **20%** of the overall score, with the remaining **80%** being based on the technical score. Therefore, it is crucial to prepare the pitch thoroughly.

Here are some key points to include:

- Present your complete solution, including the efforts made to solve the riddles.
- Discuss the challenges encountered and how each team member collaborated to implement the solution.
- Share your vision for scaling up the solution and potential additional use cases it can serve.
- Present any possible improvements that can be implemented to enhance your solution.