

Annotation Guide

Introduction

In this project, we want to evaluate the potential of Large Language Models (LLMs) to mimic Secure Shell (SSH) servers as components of honeypots in cybersecurity. This guide is designed to assist annotators in understanding their role in this research.

The goal of this annotation task is to evaluate the convincingness of responses generated by GPT-3.5 when presented with requests typically encountered by SSH servers. The annotators will compare these generated responses to those of an actual SSH server, determining their plausibility and effectiveness in mimicking real interactions. This task focuses less on the technical accuracy of the responses and more on their overall believability, considering what a human attacker might find convincing.

Annotation Task Details

Structure of the data

Each data point that needs to be annotated, is composed of a three-part format designed to simulate an SSH session interaction. This format provides context for evaluating the LLM's ability to mimic real SSH server responses accurately.

Command: This element represents a SSH command that was sent to a real SSH Server and an LLM. It is the basis for the evaluation.

True Response: This is the actual output from a real server (a debian jessie system on an x86_64 architecture) in response to the command sequence. It serves as a reference for what a typical SSH server might return but is not shown to the LLM. This expected response helps in gauging the accuracy and realism of the LLM's generated response.

Completion Response: This is the LLM-generated response based on the completion request (command and session history in the halle dataset (see below)). It is the focus for the evaluation. Although the true response serves as a benchmark, a completion response that is plausible yet different from the true response is still considered valid.

Session History (Halle Dataset): In the Halle dataset, the language model was given context by providing it with the history of commands and responses from a specific session. This means that all the commands issued during the session, along with the responses received from the SSH server (true response), were sequentially sent to the language model. This allowed the model to have access to the full conversation history up to the point of the last command. For this final command, the language model was then asked to generate a response, utilizing the provided session history as context. It is crucial to note that the provided history was that of the real SSH server (true responses; for consistency purposes), not the previously generated responses by the LLM itself.

Definition of Convincing

In conducting annotation, we adhere to the definitions provided by the Cambridge Dictionary:

- Convincing: Able to make you believe that something is true or right.
- Believe: To think that something is true, correct, or real.

Within the context of this project, the term convincing refers to the degree to which a response from an LLM to an SSH server request appears authentic and credible, as if it originated from a real SSH server. The notion of convincingness is not strictly tied to technical precision. Even if a response contains technical inaccuracies, it can still be considered convincing provided it aligns with how an SSH server would typically react under comparable circumstances. The focus lies on the response's authenticity and its ability to convincingly mislead a potential attacker into believing they are interacting with a genuine SSH server.

Binary Annotation System

For the purpose of this study, we employ a binary annotation system, where responses are classified as either 0 (Non-Convincing) or 1 (Convincing).

0 (Non-Convincing): Assign this rating to responses that fail to convincingly mimic a real SSH server interaction. This includes responses that are technically inaccurate to the point of implausibility, lack context, or would not be expected from a real SSH server.

1 (Convincing): Assign this rating to responses that convincingly mimic a real SSH server interaction, displaying a level of plausibility and coherence that could deceive someone into believing it is from an actual SSH server.

Criteria for Annotation

The following criteria should guide the decision-making process:

1. **Contextual Appropriateness:** The response should be appropriate for the request, fitting the context of typical SSH server interactions.
2. **Technical Plausibility:** While exact technical accuracy is not the primary focus, the response should not contain glaring errors that would immediately alert a technically informed attacker to its artificial nature.
3. **Consistency and Coherence:** The response must be internally consistent and coherent, adhering to the established context and not deviating into contradictory or nonsensical territory. It should logically follow from the completion request and maintain consistency with known SSH server behaviors and responses.

Detailed Examples and Edge Cases

The following examples are intended to clarify the criteria for convincingness and provide concrete instances of how to apply the binary annotation system in practice.

Example 1: Convincing Response

Completion Request:

user:
`ls -l`

True Response:

```
total 16
drwxr-xr-x 2 user group 4096 Jan 10 10:00 Documents
-rw-r--r-- 1 user group  512 Jan 10 09:58 file.txt
-rw-r--r-- 1 user group  512 Jan 10 09:58 file2.txt
-rw-r--r-- 1 user group  512 Jan 10 09:58 file3.txt
```

Completion Response:

```
total 8
drwxr-xr-x 2 user group 4096 Jan 10 10:00 Documents
-rw-r--r-- 1 user group  512 Jan 10 09:58 file.txt
```

Annotation: 1 (Convincing)

Rationale: The response accurately reflects what would be expected from an SSH server when asked to list directory contents. While the response from a real server (expected response) is slightly different, because on the real server were more files present, the generated response has the same format, includes typical file permissions, user and group information, and file sizes, making it plausible for an attacker (that does not know which files to expect on the target server).

Example 2: Non-Convincing Response

Completion Request (with history):

user:
`mkdir pictures`

SSH server:

`<empty response>`

user:

`ls -l`

True Response:

```
total 12
drwxr-xr-x 5 user group 4096 Jan 10 10:00 Documents
drwxr-xr-x 2 user group 4096 Jan 12 11:00 pictures
-rw-r--r-- 1 user group 512 Jan 10 09:58 file.txt
```

Completion Response:

```
total 12
drwxr-xr-x 5 user group 4096 Jan 10 10:00 Documents
-rw-r--r-- 1 user group 512 Jan 10 09:58 file.txt
-rw-r--r-- 1 user group 512 Jan 10 09:58 file2.txt
```

Annotation: 0 (Non-Convincing)

Rationale: The generated response fails to reflect the recent creation of the new folder as indicated in the completion request. A real SSH server, when asked to list directory contents after creating a new folder, would show the newly created folder in its output. The omission of the folder in the generated response is a significant inconsistency that would alert a technically informed attacker to the artificial nature of the interaction. This discrepancy demonstrates a lack of contextual appropriateness and technical plausibility, making the response not convincing.

Edge Case 1: Technically Inaccurate but Plausible

Completion Request:

user:

```
sha256sum important_file.txt
```

True Response:

```
e3b0c44298fc1c149afbf4c8996fb924 important_file.txt
```

Completion Response:

```
d41d8cd98f00b204e9800998ecf8427e important_file.txt
```

Annotation: 1 (Convincing)

Rationale: The generated response correctly mimics the format of a hash function output, displaying what appears to be a valid hash value followed by the file name. This demonstrates an understanding of the expected output format for a hashing command. However, the hash value provided does not match the expected value for the file, indicating a technical inaccuracy. Despite this, to an attacker without the exact hash values or without specific expectations of what the file's hash should be, this response may still appear plausible. It demonstrates the LLM's capability to generate convincing command outputs, even if the specific details are technically incorrect. This is crucial for a honeypot, as the objective is to appear authentic and engage the attacker rather than to be an exact replica of the targeted system.

Edge Case 2: Time is not accurate

Completion Request:

user:

```
cal
```

True Response:

```
February 2024
Su Mo Tu We Th Fr Sa
                1 2 3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29
```

Completion Response:

```
November 2019
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
```

Annotation: 1 (Convincing)

Rationale: While some LLMs cannot determine the time right due to the knowledge cut-off, our expected responses originate from a wide timespan of data collection. Therefore, we focus on the format of the output. If the formats match, we will neglect the discrepancy of unmatching dates and times.

Edge Case 3: Technically inaccurate and not plausible

Completion Request:

user:

```
sudo chown hduser:hadoop ${directory_path}
```

True Response:

chown: invalid user: 'hduser:hadoop'

Completion Response:

chown: changing ownership of '\${directory_path}': Operation not permitted

Annotation: 0 (Non-Convincing)

Rationale: While a different error than the expected one could indeed be acceptable, in the generated response the LLM did not resolve the value of the variable. That would not happen on a real server.

Annotation Process

For the annotation you will be provided with a google drive folder and you are supposed to add the annotations in the respective google table file. We prepared a Column “convincing” for you to fill with either value 1 or 0 - according to your assessment whether or not the generated response is believable.

We suggest beginning with the "Prague" dataset as it comprises the simplest commands, offering a straightforward introduction to what you can expect from other datasets. Afterwards, the "nl2bash" dataset could be a good next step, as it builds on the basics without the added complexity of session histories found in the "Halle" dataset. As a visual aid, we have highlighted related sessions in alternating colors. The session ID can also be used to identify which sessions belong together.

If you like, you can adjust the google sheet documents to your needs (e.g., change the column width or row height). The documents are just for you and we will retrieve your annotation results programmatically.

Please feel free to reach out to us, if you have any questions or remarks.