JPMorgan Chase & Co.

Anomaly Detection of Command Shell Sessions based on DistilBERT: Unsupervised and Supervised Approaches

Conference on Applied Machine Learning in Information Security (CAMLIS)
October 19-20, 2023, Arlington, VA

Zefang Liu, John Buford
JPMorgan Chase

Introduction

Research Backgrounds

Backgrounds:

- **Interactive command shells**, especially Unix shells, which provide a powerful interface for system administration, development, and maintenance tasks, **can be exploited by attackers** to gain unauthorized access, escalate privileges, avoid defense detection, collect sensitive data, and manipulate systems.
- Previous studies utilized various techniques for anomaly detection in command shell sessions, ranging from simple **rule-based methods** to more complex **machine learning algorithms**, rely heavily on predefined features or labeled data from security experts for training supervised models.
- Recent advances in deep learning and natural language processing, in particular **transformer-based models**, such as BERT and GPT, have the potential to enhance computer security by enabling more effective and adaptable anomaly detection systems that can learn from large-scale, diverse data sources.

Introduction

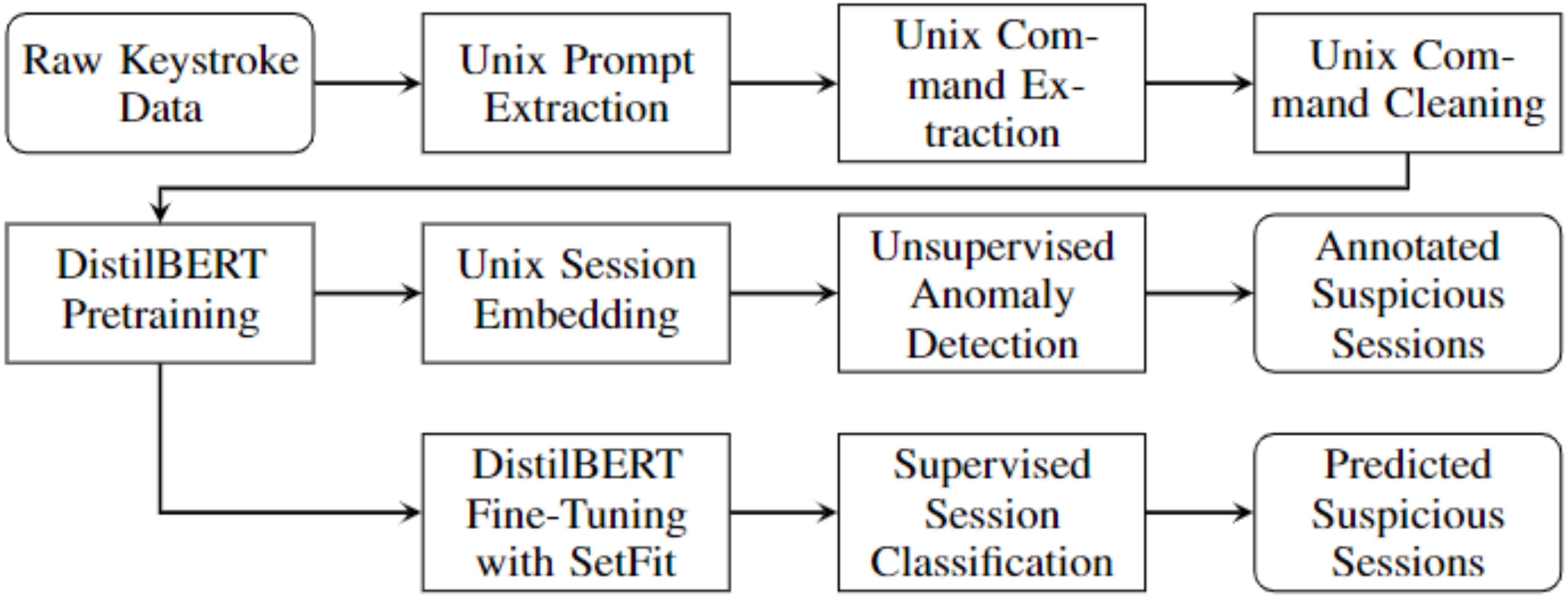
Research Motivations

Motivations:

- In enterprise production environments, access to **command shells** is treated as a privileged activity because of the potential for misuse of system commands.
- Enterprises can implement **rule-based detection** using computer security frameworks. However, due to the volume, length, and complexity of shell sessions, manual detection of outliers is impractical.
- The anomaly detection model can :
  - **Automatically identify command patterns** that are outliers with respect to the overall set of sessions, which would not be detected by the rule-based approach, and
  - **Assign anomaly scores to sessions**, where sessions with high anomaly scores can be prioritized for further investigation.

Introduction

Research Contributions


Raw Keystroke Data
Unix Prompt Extraction
Unix Command Extraction
Unix Command Cleaning
DistilBERT Pretraining
Unix Session Embedding
Unsupervised Anomaly Detection
Annotated Suspicious Sessions
DistilBERT Fine-Tuning with SetFit
Supervised Session Classification
Predicted Suspicious Sessions

Main contributions:

- We apply **a comprehensive anomaly detection framework for Unix shell sessions** based on the pretrained DistilBERT model and ensemble anomaly detectors, addressing an important problem in computer security.
- We conduct experiment and demonstrate the effectiveness of **unsupervised approach** using an ensemble method to compute anomaly scores for a large-scale enterprise dataset, enabling the identification of suspicious activity without extensive manual labeling.
- We evaluate the performances of supervised fine-tuned models on a few-shot set of labeled sessions, highlighting the adaptability and accuracy of our **supervised approach**.

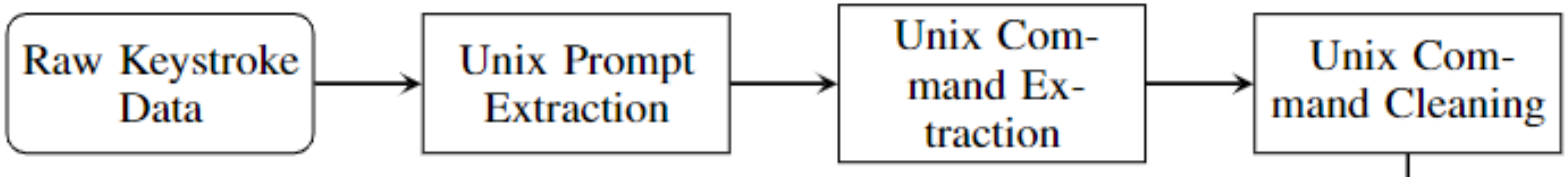
Data

Data Description and Characteristics

- Data: **a large-scale, unlabeled dataset of Unix shell commands from real operating system users**.
- The raw data used includes **90 days of Unix keystroke sessions** from **over 15,000 users**, which have about 3 million activity objects, where around 2.4 million objects are non-empty interactive sessions.
- The raw data have several data characteristics:
  - Mixed shell prompts, command inputs, and command outputs,
  - Various shell prompts across sessions and within session,
  - Truncated long command lines with varying line lengths,
  - Various command aliases across sessions,
  - Mixed background process outputs with prompts and inputs, and
  - Missed backspaces and tab keys.
- We developed heuristics to extract and clean commands from the raw data. After extracting commands and dropping duplicates, we obtain **1.15 million sessions**.

Data

Data Preprocessing



Prompt extraction:

- A list of 140 common Unix commands and a list of prompt terminal symbols ($, #, >) are created.
- For each input line, the first occurring prompt terminal symbol is located, and the following word is tested against the common command set.
- If this word is a known common command, the prompt is saved.

Command extraction:

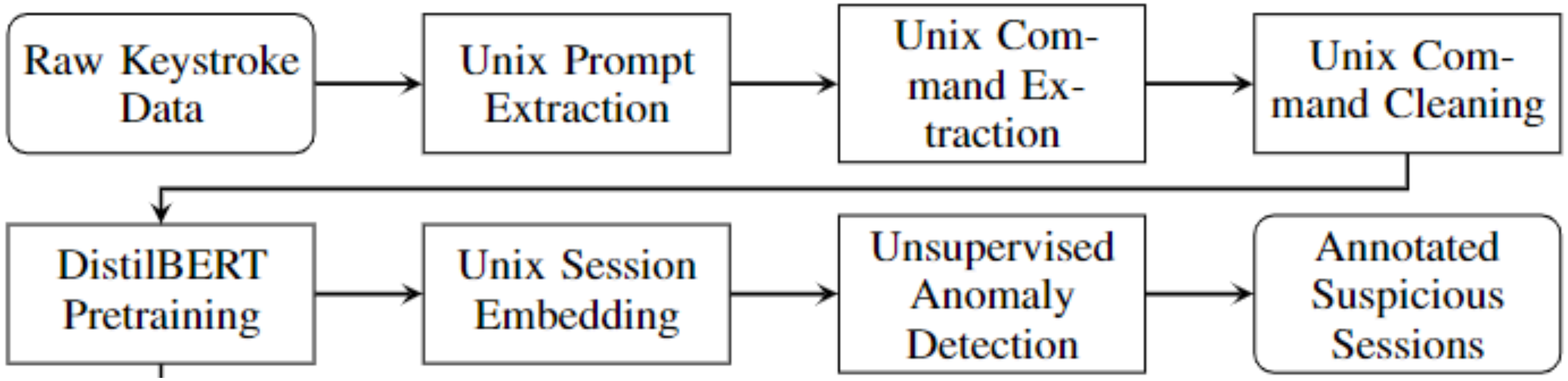
- Known prompts from this session are searched and then the command lines after the prompt are extracted.
- Text editor buffers are removed and wrapped multiple-line commands are concatenated.

Command cleaning:

- Several filters are applied:
  - Removing command lines with error messages,
  - Dropping command editing buffers and shell completions,
  - Deleting long consecutive spaces and over-repeated characters,
  - Filtering command names with regular expressions,
  - Masking numbers and special words, and
  - Cleaning cyclic commands usually generated by loops from shell scripts.

Unsupervised Approach

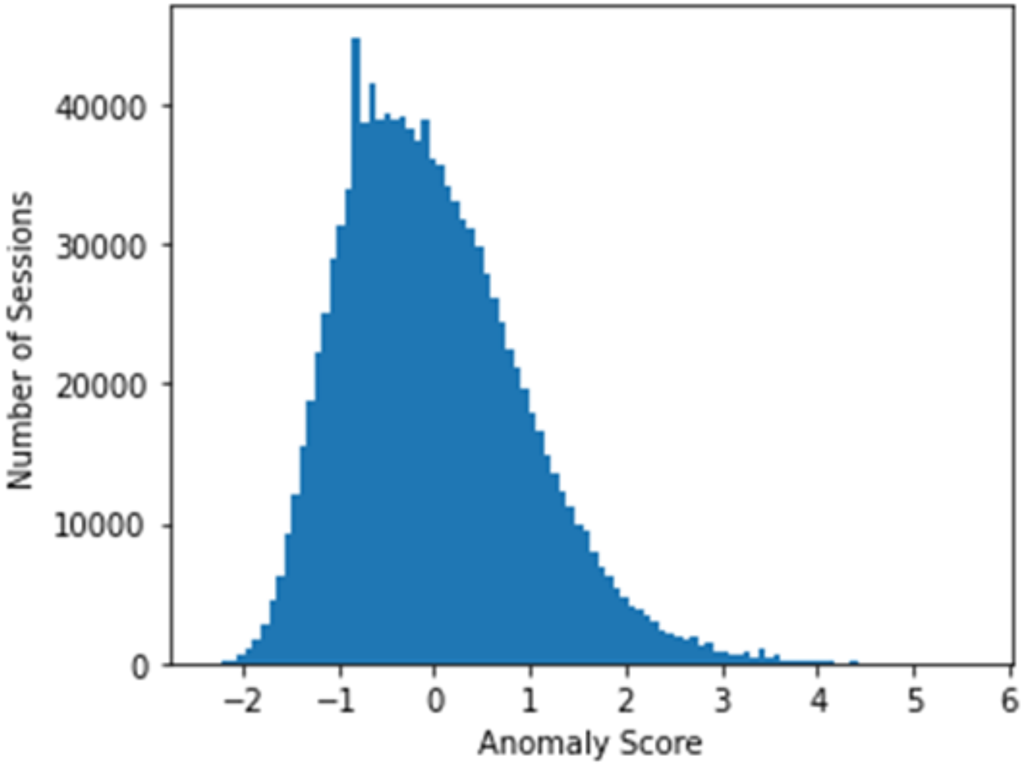
DistilBERT Pretraining

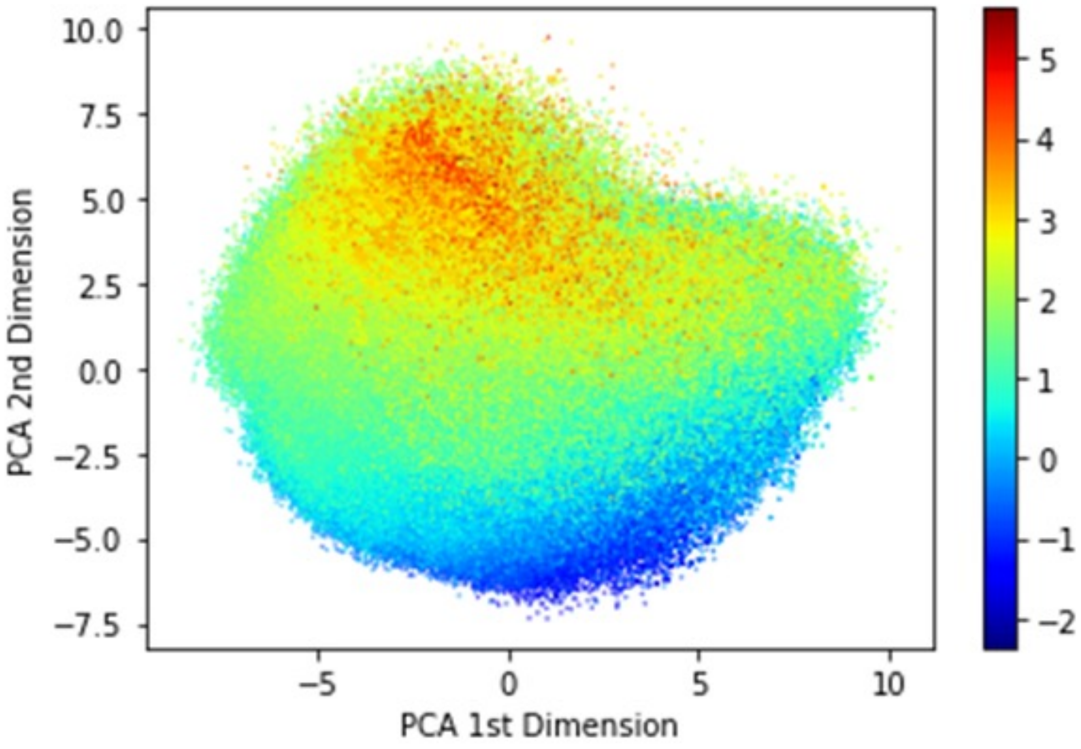


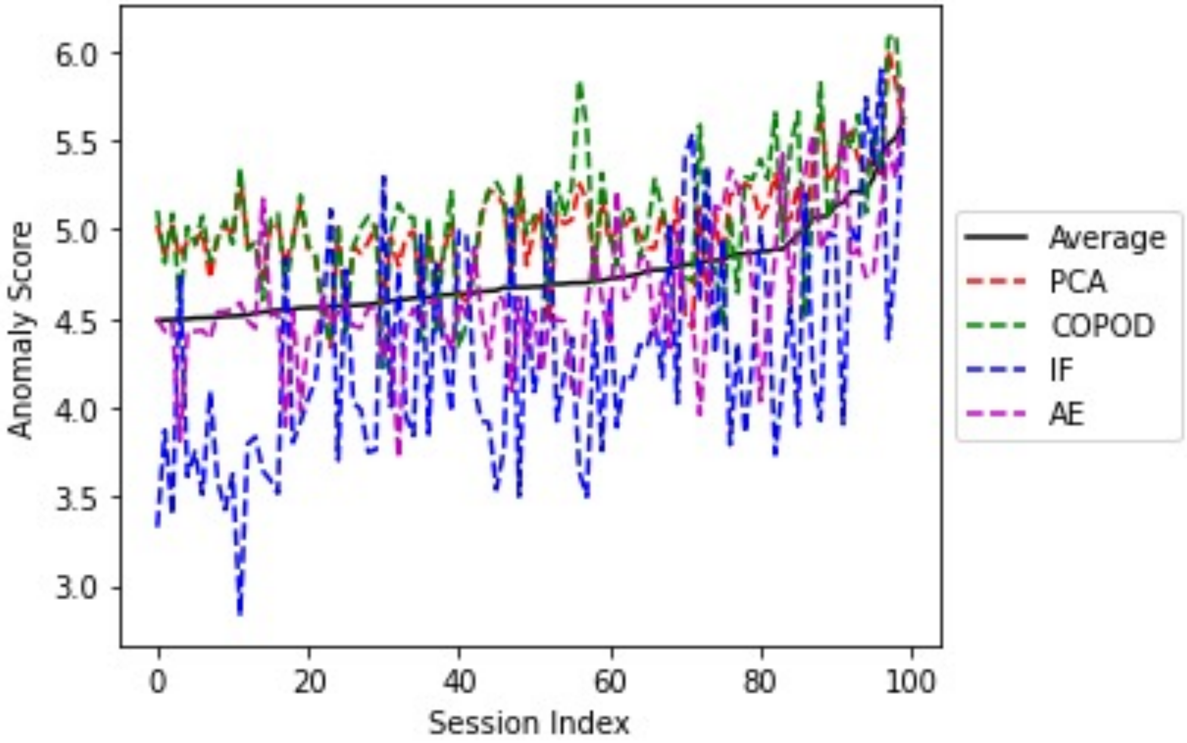
- The unsupervised approach involves **pretraining a DistilBERT on Unix shell commands** and **constructing an ensemble anomaly detector**, which was first proposed by CrowdStrike® for command lines from various platforms.
- **DistilBERT** is selected due to its balance of performance and efficiency. The **WordPiece** tokenizer with a dictionary size of 30,000 is selected and trained for tokenizing the Unix sessions.
- The DistilBERT is pretrained for the **masked language modeling (MLM)** task to capture the inherent structure and dependencies within command sequences. The last hidden states are used as the **contextual embeddings** of the Unix shell sessions.
- **Four outlier detectors**, the principal component analysis (PCA), isolation forest (IF), copula-based outlier detection (COPOD), and autoencoders (AE), are trained with the session embeddings, and their **decision scores** are normalized for each outlier detector.
- For each session, all four decision scores are averaged to get the final **anomaly score** of that session. Sessions with **high anomaly scores** are considered **outliers**, which may contain unusual command syntaxes or patterns deviant from the overall collection of sessions.

Unsupervised Approach Results

Distribution of Anomaly Scores

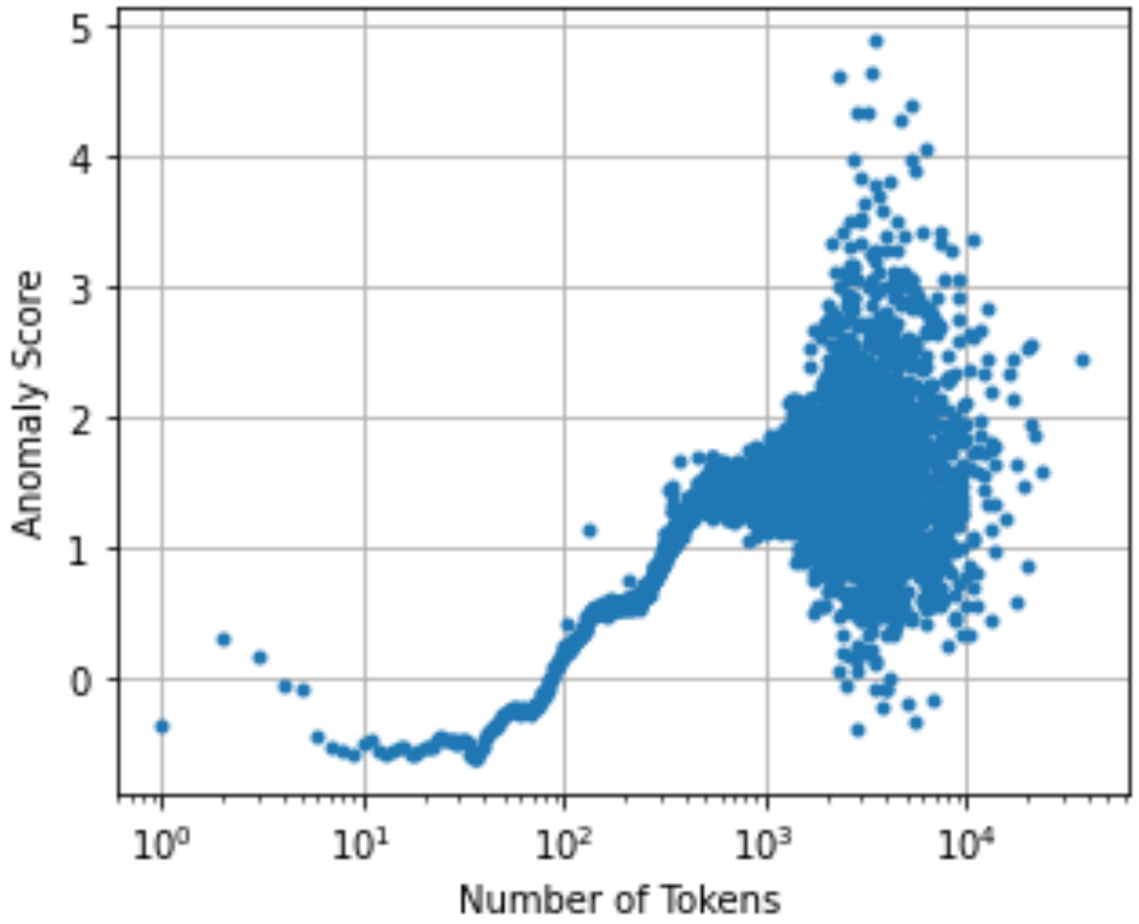


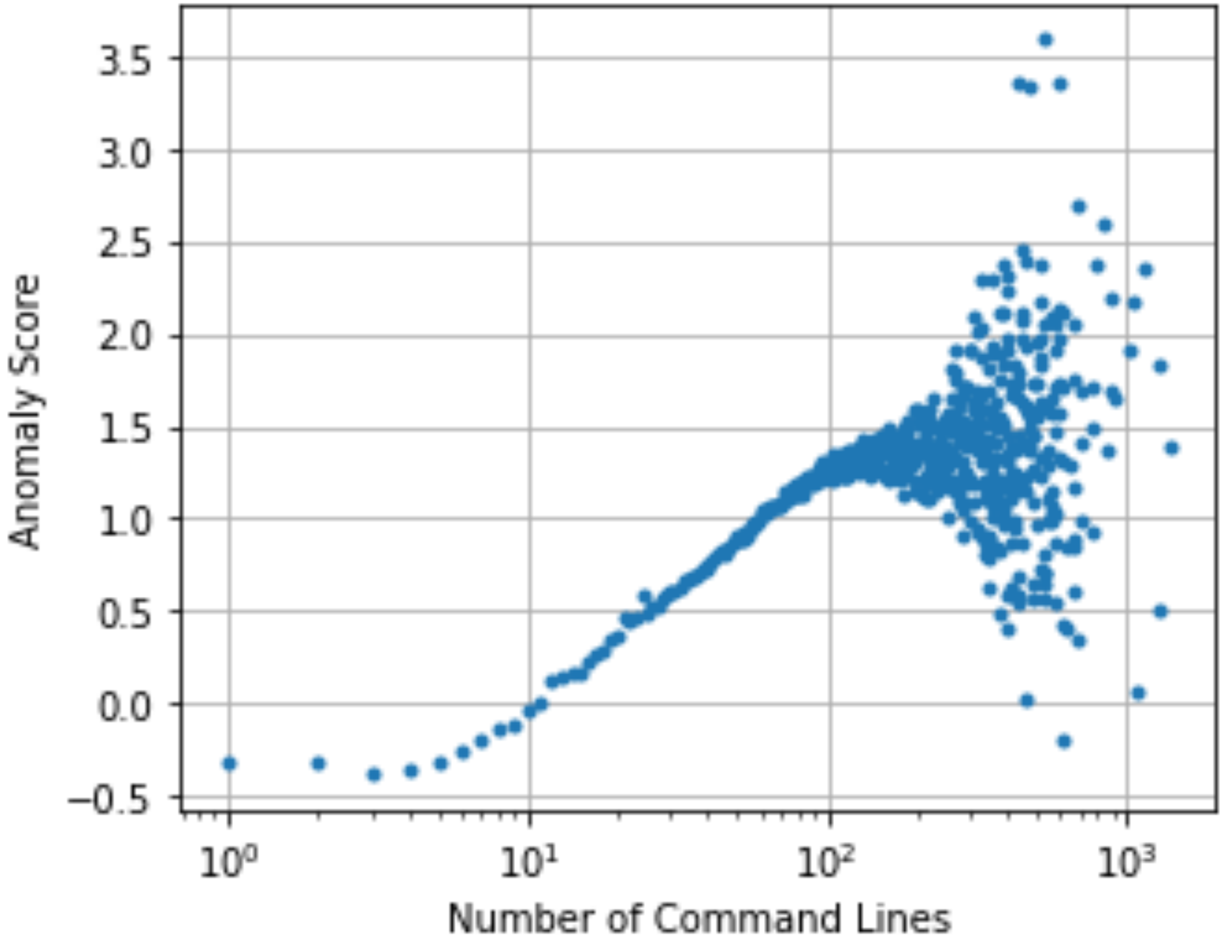




Unsupervised Approach Results

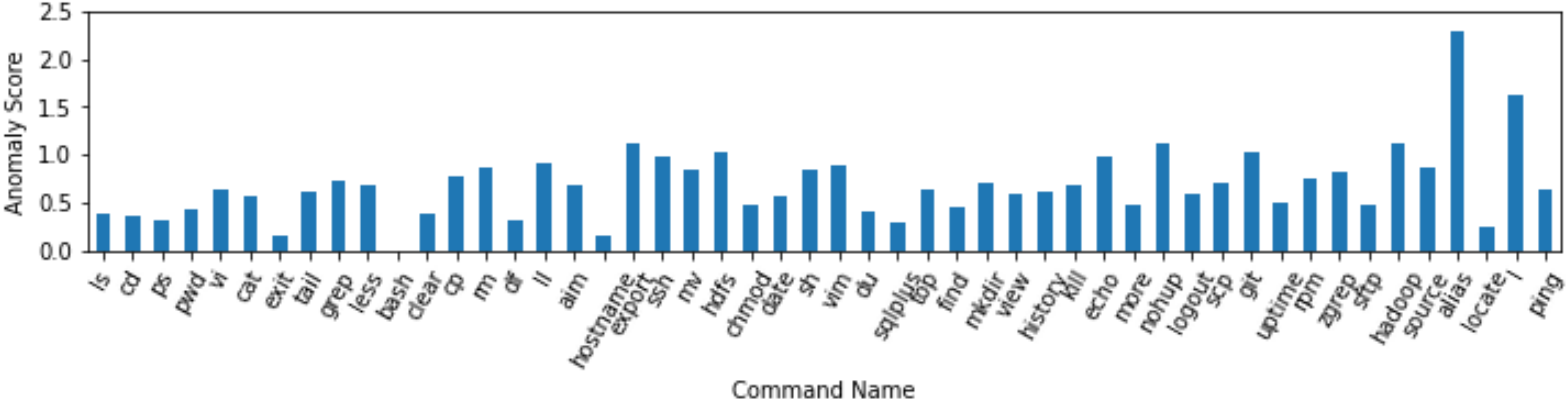
Distribution of Anomaly Scores





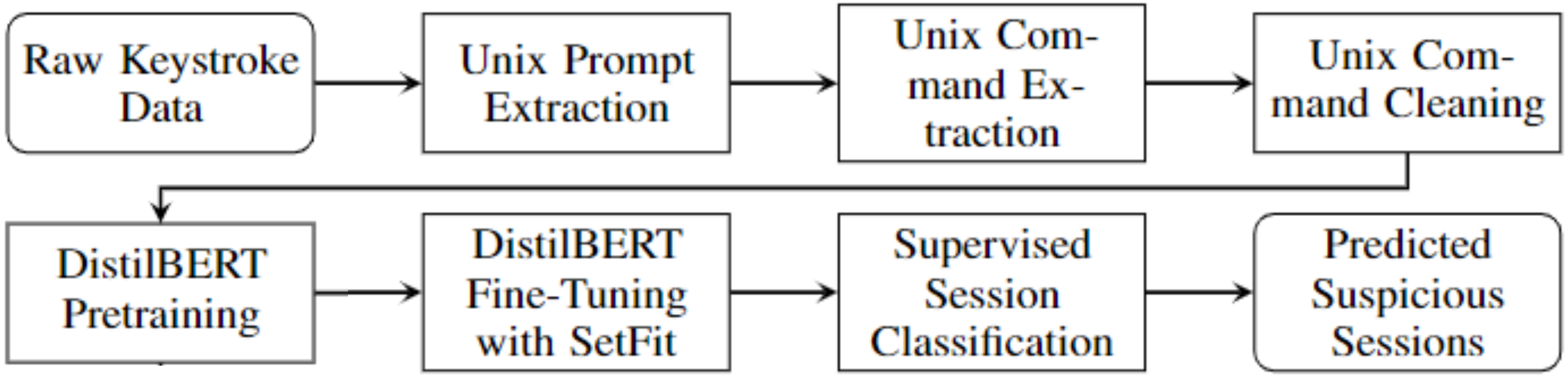
Unsupervised Approach Results

Distribution of Anomaly Scores



Supervised Approach

Pretrained Model Fine-Tuning



- The supervised part of our approach involves **fine-tuning the pretrained DistilBERT model** with labeled data to improve its performance in distinguishing between normal and suspicious command sequences as **a binary classifier**.
- We **fine-tune the pretrained DistilBERT with SetFit** (Sentence Transformer Fine-tuning), which is an efficient and prompt-free framework for few-shot fine-tuning of sentence transformers.
- We create **a table of suspicious keywords** developed based on Uptycs's work to cover MITRE ATT&CK® techniques commonly used by attackers.
- The suspicious keywords are searched in each Unix shell sessions, and those sessions with the **number of unique suspicious keywords** higher than the **threshold** are considered as **anomalies**.
- We create regular expressions to tag sessions with more **ATT&CK techniques**, which are used for session annotations.
- We evaluate the performance of the anomaly detection approach by calculating **precision, recall, and F1 score**.

Supervised Approach

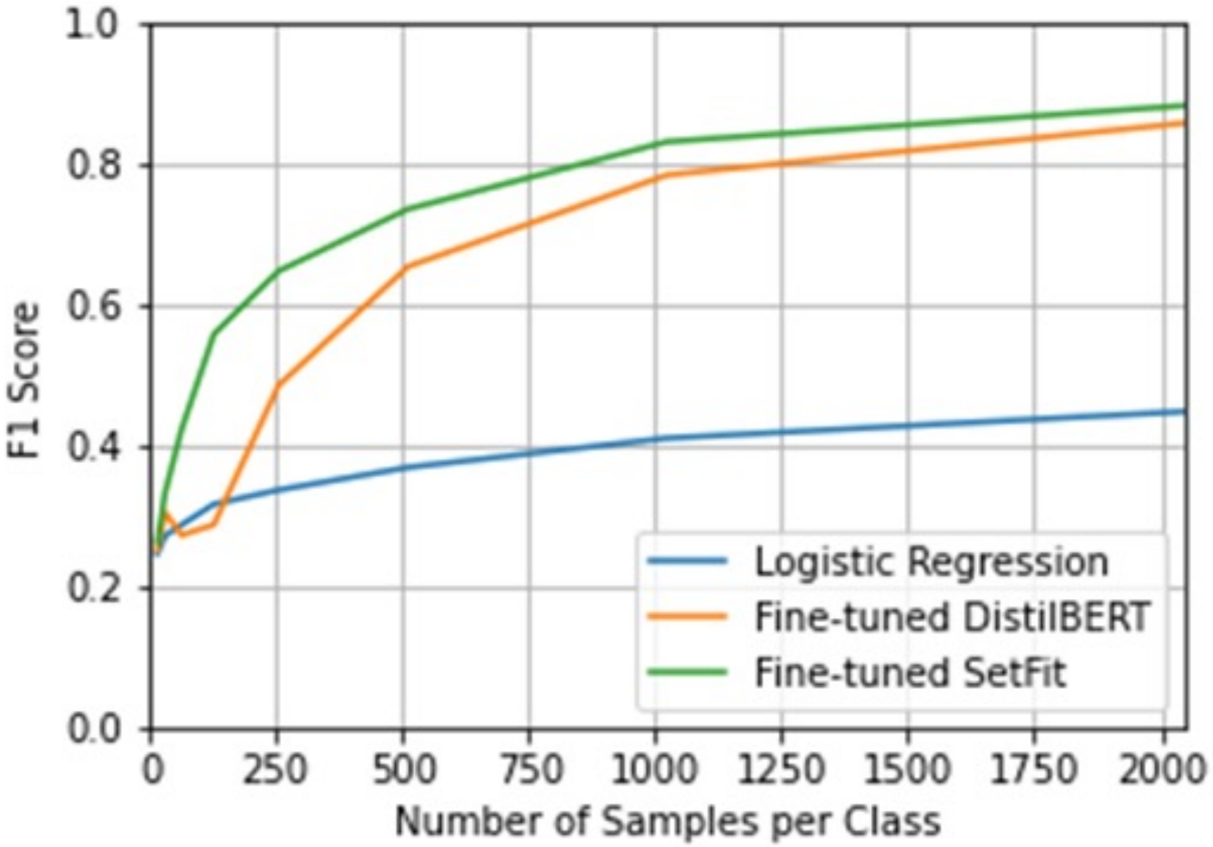
Session Labeling Rules (Examples)

ATT&CK Technique ID	ATT&CK Technique Name	Suspicious Keywords
T1018	Remote System Discovery	arp, ping, ip, hosts
T1082	System Information Discovery	df, uname, hostname, env, lspci, lscpu, lsmod, dmidecode, systeminfo
T1040	Network Sniffing	tcpdump, tshark
T1489	Service Stop	kill, pkill
T1105	Ingress Tool Transfer	curl, scp, sftp, tftp, rsync, finger, wget
T1222.002	File and Directory Permissions Modification: Linux and Mac File and Directory Permissions Modification	chown, chmod, chgrp, chattr
T1003.008	OS Credential Dumping: /etc/passwd and /etc/shadow	passwd, shadow
T1070.003	Indicator Removal: Clear Command History	.bash_history, HISTFILE, HISTFILESIZE

Supervised Approach Results

Evaluation Results of Supervised Models

Class	Number of Unique Suspicious Keywords	Number of Samples	Training Set (90%)	Testing Set (10%)
Normal	= 0	790,363	711,327	79,036
Abnormal	>= 3	28,413	25,571	2,842
Abstaine d (no label)	In between	335,322	-	-
Total	-	1,154,098	736,898	81,878



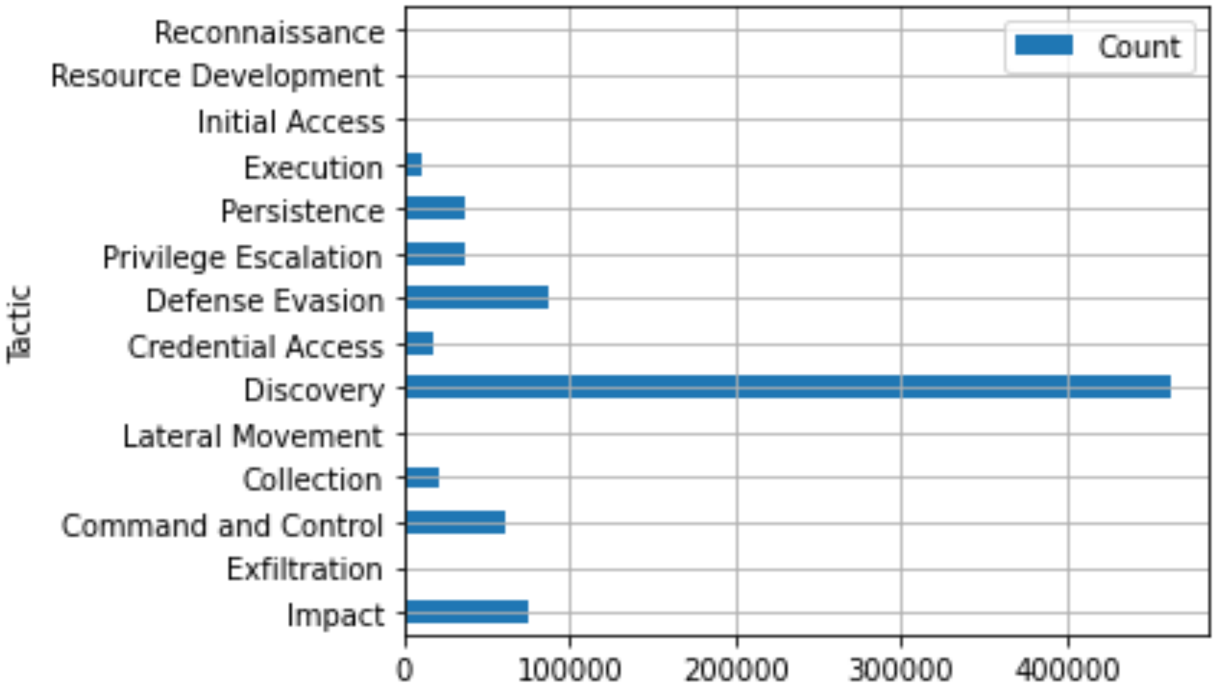
Supervised Approach Results

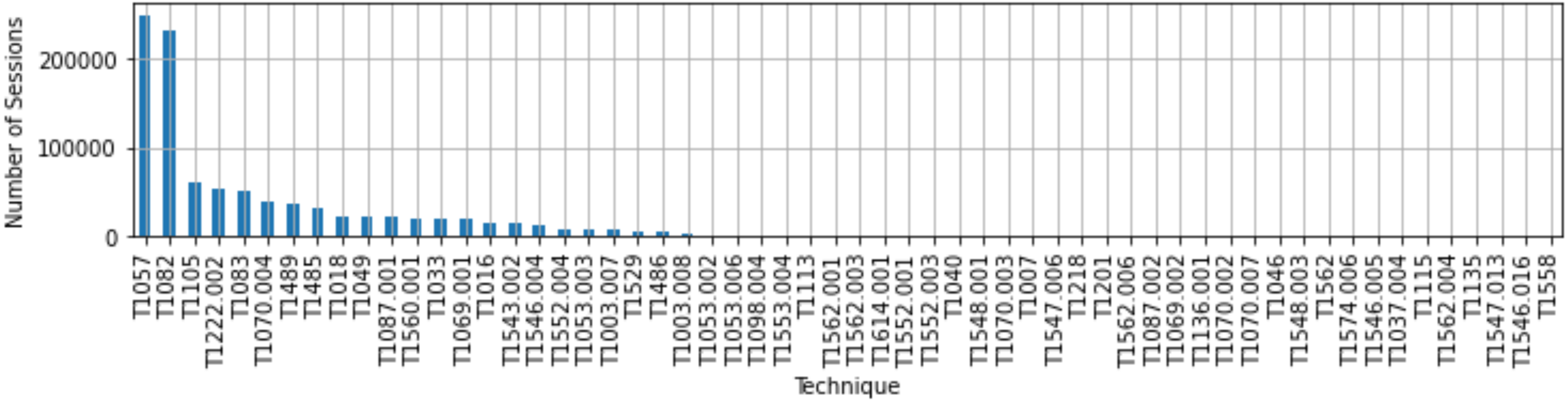
Evaluation Results of Supervised Models

Model	**Logistic Regression**			**Fine-tuned DistilBERT**			**Fine-tuned DistilBERT with SetFit**		
Number of Samples per Class	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**
16	0.1464	0.7860	0.2454	0.1632	0.5578	0.2513	0.1569	0.8287	0.2622
32	0.1625	0.8248	0.2711	0.1995	0.6977	0.3036	0.2059	0.8930	0.3331
64	0.1713	0.8754	0.2862	0.1625	0.8418	0.2716	0.2712	0.9484	0.4210
128	0.1922	0.8849	0.3155	0.1703	0.9098	0.2864	0.3909	0.9758	0.5563
256	0.2070	0.8890	0.3356	0.3230	0.9663	0.4840	0.4819	0.9850	0.6459
512	0.2308	0.9027	0.3676	0.4900	0.9774	0.6524	0.5845	0.9866	0.7337
1024	0.2631	0.9188	0.4090	0.6483	0.9854	0.7819	0.7134	**0.9900**	0.8290
2048	0.2944	0.9267	0.4467	0.7534	0.9899	0.8555	**0.7934**	0.9894	**0.8802**

Supervised Approach Results

Session Numbers of MITRE ATT&CK® Tactics and Techniques





Session Annotation Examples

Sessions Annotated with Suspicious Keywords and Techniques

Example: Remote command execution of transient web server with potential for data exfiltration.
Details: Line 2: launch transient web server on remote host.
Line 3: terminate the server.
Line 4 and 6: MITRE ATT&CK tags inserted by processing pipeline.
Line 5: transfer data from web server using wget.

Activity id = *1e1BD9. **Anomaly score = 1.8919.** Suspicious keywords = [kill: 3, wget: 21]

```
1    <lines removed>
2    salt "WH" cmd.run "python -m SimpleHTTPServer # --directory /sqldata/ms_backups/"
     bg=trues/WH_test_db_FU
3    salt "WH" cmd.run "ps aux | grep '[S]impleHTTPServer #' | awk '{print $#}' |xargs kill –9
     "/WH_test_db_FUWH:
4    -> [T1057: Process Discovery, T1489: Service Stop]
5    salt "WH" cmd.run "cd /sqldata/dbmigration;wget http://<host:port>//sqldata/ms_backups/WH_test_db_FU
6    -> [T1105: Ingress Tool Transfer]
7    <lines removed>
```

Session Annotation Examples

Sessions Annotated with Suspicious Keywords and Techniques

Example: Potential data exfiltration and credential exposure subject to discovery via process discovery.

Activity id = *1c01C8. **Anomaly score = 1.9754**. Suspicious keywords = [curl: 12]

```
<lines removed>
curl -T server_support.tar.gz -u<username>:<plaintext_credentials> <externalhost> /dropzone/uploads
-> [T1105: Ingress Tool Transfer]
<lines removed>
```

Session Annotation Examples

Sessions Annotated with Suspicious Keywords and Techniques

<table>
<tr><td colspan="2">Example: Disk clear and boot load configuration changes.
Details: Lines 1, 7, 10 omitted for brevity. Line 3 and 5 are automatic annotations added by pipeline.
Line 2: remote command to check system details. Line 4: remote command to clear disk prior to install.
Line 6: restart Hadoop monitoring agent. Line 8, 9: modify boot loader.</td></tr>
<tr><td colspan="2">Activity id = *b41A0E. <b>Anomaly score = 3.1271.</b> Suspicious keywords = [chmod: 2, df: 1, wget: 1]</td></tr>
<tr><td>1</td><td>< lines removed ></td></tr>
<tr><td>2</td><td>ansible all -i <INVENTORY> -m shell -a "uptime;grep Start /etc/INSTALL_CLASS;cat /etc/redhat-release" -o</td></tr>
<tr><td>3</td><td>-> [T1082: System Information Discovery]</td></tr>
<tr><td>4</td><td>ansible all -i <INVENTORY> -m shell -a "cd /root;chmod HFF diskwipe.sh;./diskwipe.sh" -b</td></tr>
<tr><td>5</td><td>-> [T1222.002: File and Directory Permissions Modification - Linux and Mac File and Directory Permissions Mod]</td></tr>
<tr><td>6</td><td>ansible all -i <INVENTORY> -m shell -a "/sbin/service ambari-agent restart" -become -b</td></tr>
<tr><td>7</td><td><lines removed></td></tr>
<tr><td>8</td><td>ansible all -i <INVENTORY> -m shell -a "cd /boot/grub#;cp -p grub.cfg grub.cfg.bkp" -b</td></tr>
<tr><td>9</td><td>ansible all -i <INVENTORY> -m shell -a "/sbin/grubby --args=transparent_hugepage=never --update-kernel=ALL " -b</td></tr>
<tr><td>10</td><td><lines removed></td></tr>
</table>

Conclusions

Research Summary

- **Detection of anomalies for interactive command shells** is needed as a cybersecurity safeguard because privileged access at the shell level provides the opportunity for a range of attacks that threaten critical enterprise infrastructure, data, and services.
- We presented the **first published results on keystroke anomaly detection using an enterprise-scale dataset** captured from production systems over a 90-day period with 1.15 million sessions and over 15,000 users.
- We presented the **first experimental results of using the transformer model**, specifically DistilBERT, **for keystroke log anomaly detection of Unix shells**, in both **unsupervised and supervised approaches**.
- We tagged each session using two existing schemes, the **MITRE ATT&CK® techniques** and **suspicious keywords**, where Unix shell sessions with high anomaly scores were then cross-checked with the tags as part of validating the utility of the anomaly model for operations uses.

Thanks for Listening!