Unveiling Evolving Threats: A Data Analysis for Next-Generation Honeypot Development

Ziyang Wang[1,2], Shichao Lv[1,2†], Haining Wang[3], Jianzhou You[4], Shuoyang Liu[1,2],
Tianwei Yuan[1,2], Xiao Lu[1,2], Limin Sun[1,2†]
[1] Institute of Information Engineering, CAS [2] School of Cyber Security, University of Chinese Academy of Sciences
[3] Virginia Tech [4] Sangfor Technologies Inc.

Abstract—**Honeypots act as a powerful security mechanism that diverts malicious actors from production systems while providing valuable insights into adversarial behaviors. Yet, the absence of a high-quality honeypot dataset has long impeded robust benchmarking and restricted the employment of advanced AI-driven honeypot defenses. In this work, we address these limitations by constructing a comprehensive shell request-response dataset from Cowrie honeypots. Such a dataset not only facilitates thorough, in-depth honeypot evaluations but also furnishes an essential research foundation for AI-based honeypot development. We analyzed tens of thousands of shell sessions collected during two distinct time frames. (2021–2022 and 2024). By systematically examining command-level usage, session behaviors, and tactics under the MITRE ATT&CK framework, we identified major shifts in adversary techniques, including mounting command complexity, shorter but more targeted infiltration sessions, a more balanced and diverse range of attack methods, and an intensified focus on circumventing defensive measures. These observations emphasize the evolving nature of shell-based intrusions and underscore the necessity for ongoing experimentation and iterative improvements in honeypot design. Through the collection and analysis of this dataset, our work highlights emerging threats in shell defense while also establishing a robust data foundation for the future development of AI-driven honeypots.**

I. Introduction

The cybersecurity landscape evolves constantly as adversaries relentlessly refine attack tactics to exploit emerging vulnerabilities. Among defensive mechanisms, honeypots serve as a useful tool for diverting attackers and uncovering their attack behaviors by emulating vulnerable services without risking actual information systems [2], [9], [12], [19]. However, the development of modern honeypot systems, particularly those leveraging artificial intelligence (AI), is limited by the lack of open-source, standardized, high-fidelity, high-interaction request-response datasets that accurately capture evolving threat scenarios [1], [21], [28]. Existing datasets often lack robust evaluations of the diversity and balance of attack techniques or suffer from outdated information, thereby reducing their usefulness for honeypot evaluation, threat analyses, and the update of defense mechanisms [3], [16].

To address these limitations, this work constructs a novel shell command request-response dataset derived from Cowrie honeypot deployments across two distinct time periods (2021–2022 and 2024). Unlike prior efforts that focus on isolated attack logs or passive traffic capture [3], [16], we systematically replay captured attacks within meticulously engineered environments that emulate production systems, ensuring authenticity and interactivity. Furthermore, we map over 10,000 post-login shell sessions to the MITRE Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) framework [22], labeling each interaction with its corresponding tactics and techniques. This granular annotation enables two critical advancements: (1) longitudinal analysis of attacker Tactics, Techniques, and Procedure (TTP) evolution and (2) systematic evaluation of honeypot response efficacy against specific adversarial techniques.

Our comparative analysis between two periods reveals three key trends in shell-based attacks: (1) Quantitative examination demonstrates a marked increase in the complexity and diversity of shell commands, with attackers increasingly constructing novel, sophisticated, multi-stage complex commands that integrate diverse shell tools. (2) The significant rise in the proportion of short-duration shell sessions indicates a strategic shift toward precision and efficient environment sensing, reflecting attackers' growing awareness of traditional honeypot detection capabilities. (3) MITER ATT&CK techniques display heightened diversity and a more balanced distribution, featuring a denser integration of technical combinations alongside a pronounced preference for defense evasion tactics. These findings dismantle the long-held assumption that automated shell-based attacks are static and reveal critical gaps in the design of programming-based honeypots.

The main contributions of this paper lie in three aspects. (1) We created an open, ATT&CK annotated dataset of shell request-response interactions, which will be open-source for training and evaluation in AI-driven honeypot development. (2) We conducted a longitudinal analysis of attacker behavior evolution and identified significant shifts in adversary techniques. These include increasing command complexity, shorter yet more targeted infiltration sessions, a more balanced and diverse array of attack methods, and a pronounced focus on evading defensive measures. (3) We offered actionable recommendations to enhance shell defense, assisting defenders in adapting to evolving shell attack scenarios.

The rest of this paper is structured as follows. Section 2 provides background on shell attacks and honeypots, AI-

[†] These authors are Co-Corresponding Authors. This project is supported by National Natural Science Foundation of China (Grant No.92467201)

driven honeypot challenges, and shell attack analysis. Section 3 details shell request-response dataset construction, including command sourcing, preprocessing of attack data, and response collection. Section 4 presents a time-based comparison analysis of the shell attack situation, including new attack vectors, quantitative analysis on command and session dimensions, and a strategy situation based on MITRE ATT&CK. Section 5 discusses the implications for shell defense recommendations. Section 6 outlines the limitations and potential future work, and finally, Section 7 concludes the paper.

II. Related work

A. Shell Attacks and Honeypots

Shell-based services—such as SSH, Telnet, and command terminals—are at the core of modern IT infrastructures, providing essential remote administration capabilities and powering critical business processes [4], [20]. However, their widespread deployment and ease of access make them an attractive entry point for adversaries to exploit [10], [15], [19], [29]. Attackers frequently leverage these services by executing malicious shell commands, thus gaining the ability to move laterally within networks, escalate privileges, exfiltrate data, or install persistent backdoors. To defend against these threats, honeypots [2], [6], [9], [12], [18], [19], [24]–[26], [30] have been widely used as a deception mechanism that emulate real shell services to attract and engage attackers, while monitoring their tactics and techniques in detail. One widely adopted solution is Cowrie [14], an open-source honeypot specifically designed to emulate SSH and Telnet connections, supports detailed session logging, command tracing, and file interaction monitoring, making it a valuable tool for understanding real-world attack patterns and adapting defensive strategies accordingly. In this paper, we leverage data collected by Cowrie to analyze the shell attack landscape. By examining these honeypot insights, defenders can enhance their detection methodologies, proactively address emerging threats, and strengthen their overall defensive posture.

B. AI-Based Honeypots Training and the Need for Request-Response Datasets

Contemporary honeypot design strategies fall into three categories: real-system-based, programmatic, and AI-driven approaches. Real-system-based honeypots deploy actual vulnerable systems to attract attackers, offering high realism but posing significant rigidity in the internal environments [2], [25], [27]. Programmatic honeypots, emulate services through predefined scripts, improving flexibility but struggling to mimic complex, dynamic interactions [1], [13], [14]. In contrast, AI-driven honeypots leverage machine learning to dynamically adapt to attacker behaviors, enabling context-aware responses, reducing manual configuration overhead, and actively engaging attackers [21], [28].

While AI-based designs show promise, current implementations remain limited. Small-scale models, such as lightweight neural networks, have been applied to simple application simulations like industrial control system (ICS) protocol, where syntax is simple and limited training data is sufficient [11]. However, in shell service emulation, a domain characterized by open-ended command structures, diverse parameters, and adversarial creativity [8], such models fail to generate convincing interactions without high-quality, large-scale datasets. Consequently, researchers have turned to large language models (LLMs) like GPT-4 to simulate shell behaviors, relying on their broad linguistic knowledge to handle unpredictable inputs [28]. Yet, LLM-powered honeypots face challenges: high computational costs, latency in real-time response generation, and vulnerability to prompt injection attacks [28]. These challenges highlight the need for high-quality request-response datasets, which can enable smaller, more expert models to achieve a level of realism close to that of LLM-driven solutions without incurring the same overhead [5], [23]. By refining AI models through comprehensive request-response data, we can pave the way for lighter, faster, and more cost-effective AI-based honeypots, ushering in a new era of intelligent defensive strategies.

C. Shell Attack Situation Analysis

Accurate situational analysis of shell-based attacks is critical for enhancing defense mechanisms. Unlike broader network-level attack analysis, which focuses on initial intrusion attempts such as port scanning and brute-forcing [7], post-login shell command analysis provides granular insights into attacker behaviors after gaining access to a shell service. By examining command sequences such as malicious file execution, privilege escalation, lateral movement, and data exfiltration, researchers can uncover the intent, tactics, and techniques of adversaries, enabling the development of more effective defense strategies [8], [17].

As a framework covering tactics and techniques observed in real-world attacks, ATT&CK provides a framework for mapping shell-based activities [22]. In this work, we analyze shell attacks at two levels: tactics and techniques. Tactics represent the adversary's high-level goal such as credential access, while techniques describe the specific methods used to achieve these goals, such as credential dumping for obtaining account information. By translating raw shell commands into ATT&CK techniques, researchers can systematically analyze attack patterns, identify trends, and provide extra defensive suggestions, bridging the gap between raw data and actionable intelligence. In this work, we employ the MITRE ATT&CK framework to systematically compare shell attack data spanning 2021–2022 with more recent data from 2024, uncovering shell attack dynamics and improving suggestions for honeypot design and threat mitigation.

III. Open Dataset Creation

This section delineates the construction of our dataset, detailing the origins of attack commands, the preprocessing of attack data, and the methodologies for collecting and processing system responses.

A. Source of Attack Commands

To construct a comprehensive and extensive dataset, we utilized attack commands captured by the Cowrie honeypot system in recent years, specifically from two periods: cowrie-20210406-20210611 [3] and cowrie-20220609-20220704 [16]. Additionally, to incorporate more recent attack trends, we deployed Cowrie honeypots on the Internet for three months in 2024. Since only sessions that successfully authenticate have meaningful command interactions, and given that the same attacker might repeatedly attack, we segmented the sessions based on the IP address of the attackers. Each set of sessions from the same attacker post-authentication is considered as one effective attack session. The details of these datasets, including the volume of captured attack commands and effective sessions, are summarized in Table I.

B. Preprocessing of Attack Data

In the original dataset, several data collection errors arose due to inherent limitations in the Cowrie honeypot. To mitigate the potential adverse effects of these errors on subsequent data analysis, we addressed these issues, with focus on three key aspects:

a) Incorrect Segmentation of Request Commands: The Cowrie honeypot sometimes struggles to correctly parse shell scripts inputted into the terminal when they are segmented improperly based on the Enter key. This results in compromised data integrity and accuracy. We applied a manual repair method by identifying feature characters within the shell scripts, allowing for proper segmentation and precise parsing.

b) Redundant Echo Information: This includes password input prompts during login or user privilege elevation, echoes from download commands, and error prompts for incorrect command parameters. Such information, initially recorded as commands, does not constitute valid command data. We implemented filtering measures to ensure that only legitimate command data is retained, thereby improving the dataset's quality and analytical robustness.

c) Repeated Requests by a Same Attacker: In real-world attack scenarios, some attackers configure scripts to automatically send repeated session requests to target IPs at scheduled intervals. To prevent the distortion of data analysis by the excessive number of repeated attacks from a same attacker, which could skew the representation of commonly used attack techniques, we implemented a filtering process. This process identifies and removes duplicate requests originating from the same attacker, thereby preserving the integrity and accuracy of the dataset's representation of diverse attack methods.

C. System Response Collection and Processing

Given the frequent oversight in previous research regarding the collection of attack responses, we completed the assembly of request-response pairs to enhance our dataset.

1) Session-Based Command Grouping: Recognizing the potential impact of command order within the same attack sequence on system outputs, we grouped related attack commands into sessions. By replaying these sessions in a real system environment, we can generate comprehensive request-response pairs.

2) System Configuration Strategy: It is imperative to emphasize that the system type used for replaying attacks must match the configuration of the original honeypot. This consistency is crucial for ensuring the authenticity of the collected responses and their relevance to the dataset. Building on this foundation, we adopt three strategies to enhance the quality of the collected system response messages:

1. System Configuration Enhancements: We enhanced system configurations, such as incorporating high-performance graphics cards.

2. State Engineering: To satisfy specific attack queries, we modified system states, for example, by adding processes with mining-related keywords to fulfill commands aimed at mining activities.

3. Tool Installation: Installing tools required by attackers, such as net-tools, to facilitate accurate reproduction of attack scenarios.

3) Handling File Downloads: In scenarios involving file download commands, the expiration of file resources presents a challenge. To address this, we replaced download links in the commands with links to artificially created malicious executable files, ensuring successful response content collection.

4) Continuous Commands: For commands used for real-time monitoring and displaying system resource usage (e.g., 'top'), which run indefinitely until manually terminated, we set execution time limits. This approach allows us to capture response data within a predefined time frame, ensuring the automated collection of command responses for such continuous commands.

IV. Comparison and Analysis of the Dataset

This section compares attack datasets collected by Cowrie, a shell-service honeypot, during two periods (2021-2022 and 2024). We focus on emergent system commands, novel attacker behaviors, quantitative metrics related to command usage and sessions, and the evolution of attacker strategies. These findings yield insights into shifting cyber-attack patterns and highlight emerging challenges to established security measures.

A. New Vector of Attack Activity

A comparative analysis of attack commands between 2021–2022 and 2024 not only reveals a broader and more complex array of shell-based attack commands but also highlights the emergence of novel tactics. These findings indicate a trend toward stealthier, more automated, and increasingly sophisticated attack methods capable of bypassing traditional defense mechanisms.

1) Diversification of Shell Commands: Over time, attackers have expanded their repertoire of commands for Shell intrusions, incorporating a more extensive range of techniques for information gathering, command execution, file manipulation, system permission modifications, and trace cleaning. This evolution reflects a strategic shift towards more sophisticated

TABLE I
ATTACK DATA SET OVERVIEW

Data Set	Capture Method	Time Span	Number of Attacks	Effective Attack Sessions
cowrie-2024	Cowrie honeypot	2024/03/01-2024/06/26	3,914,173	6,658
cowrie-2021,2022	Cowrie honeypot	2021/04/06-2021/06/11; 2022/06/09-2022/07/04	5,099,153	5,365

TABLE II
NOVEL ATTACK COMMANDS OBSERVED IN 2024

Category	Command	Description
Gather Information	whoami	Display current user identity
Gather Information	hostname	View or set the system's hostname
Gather Information	la	Display file information of certain dir
Execute Command	screen	Start window manager for CLI
Execute Command	more	Pager program for viewing long files
Execute Command	chsh	Change the current shell
Execute Command	perl	Execute Perl scripts
Write File	touch	Create empty file
Manage Permission	chattr	Change file or directory attribute flags
Manage Permission	openssl	Encryption tools for generating keys
Manage Permission	lockr	Change file or directory attribute flags
Clean Trace	unset	Delete environment variables/functions
Clean Trace	export	Manage environment variables

and covert methods, necessitating that defense mechanisms should further expand their knowledge bases and detection rules. Table II summarizes the key attack commands observed in the 2024 dataset compared to those from 2021-2022.

2) Novel Attack Behaviors: We identified several novel attack behaviors, which highlight the increasing complexity and sophistication of modern cyber-attacks. Table III details these novel attack behaviors observed in the 2024 dataset. These behaviors leverage advanced techniques for information gathering, system permission modifications, trace cleaning, and establishing persistence, often executed with greater attack effectiveness and stealth compared to earlier attack patterns. First, attackers have broadened their information-gathering methods to inspect disk permissions, network information, system binary files, running processes, and GPU capabilities. In addition, creating new user accounts, changing user privileges, altering firewall rules, establishing backdoors, and terminating security processes all contribute to persistent access. Finally, the attackers further enhance their attack stealthiness by initiating new shells, deleting command histories, and clearing log files, thereby making detection and forensic analysis significantly more challenging.

B. Commands Quantitative Analysis

We present a detailed quantitative analysis of attack commands and sessions, comparing data from 2021-2022 and 2024. The goal is to identify significant trends and shifts in attackers' strategies over time by analyzing the dimensions of commands and sessions.

We categorize shell commands by extracting essential elements, including key commands, parameters, and command symbols such as pipes and redirections. This extraction enables us to formulate a command pattern for each sequence, exemplified by ”cat {param1} |grep {param2} |wc -l,” where param1 and param2 represent variable parameters in the initial command. By converting numerous commands into simplified command patterns, we capture identical command structures and intended attack strategies, thus facilitating a systematic analysis of command line operations and their potential implications.

1) Evolution of High-Frequency Attack Command Tactics: We analyze the key changes and trends in high-frequency attack commands between 2021-2022 and 2024, as detailed in Table IV. The comparative analysis reveals a significant evolution in adversary tactics across multiple stages of the attack lifecycle. Attackers have adapted their strategies to enhance their chances of success, maintain long-term control over compromised systems, and evade detection by employing increasingly complex commands, prioritizing persistence mechanisms, and adopting more aggressive defense evasion techniques.

Increased Command Complexity: The high-frequency commands in 2024 are significantly more complex than those in 2021-2022. For example, in 2024, the top 20 commands are predominantly complex commands, which integrate multiple simple commands to change directories and modify file attributes, such as ”cd param1; chattr -ia param2; lockr -ia param3”. Conversely, the top commands in 2021-2022 are predominantly simpler commands or complex commands with relatively simple structures, such as ”uname”. The increased complexity of these commands represents a tactical evolution designed to thwart detection and boost attack efficacy, thereby heightening challenges for defenders.

Discovery tactics play a crucial role in the initial stages of a cyber attack, enabling adversaries to gain knowledge about the targeted system and its internal network. These techniques allow attackers to observe the environment and identify potential avenues for exploitation. A comparative analysis of high-frequency attack commands between 2021-2022 and 2024 reveals a notable shift in the discovery tactics employed by adversaries. In 2021-2022, commands focused on gathering system information were more prevalent, including user information (ranks 1, 8), CPU details (ranks 4, 5, 10, 11, 13), file information (ranks 6, 10), memory usage (rank 7), and system task status (ranks 9, 12). However, by 2024, while these information-gathering techniques remain present, their emphasis has decreased. User information (ranks 3, 12, 14), CPU details (ranks 9, 10, 17, 20), file information (ranks 13, 15), memory usage (ranks 11, 19), and system task status (rank 18) are still observed, but adversaries now appear to prioritize persistence tactics, as evidenced by the top-ranking commands involving SSH key manipulation and user account creation (ranks 1, 2, 4). This shift in the threat landscape suggests that adversaries are no longer content with passive

TABLE III
NEW ATTACK BEHAVIORS

Category	Intent	Behavior
Information Gathering	Query disk permissions	Creates a specifically sized empty file for testing or as a placeholder
	Check disk space	Views available disk space
	Obtain network information	Attempts to get the host's public IP address
	Read common system binaries	Reads binaries like ls, /bin/echo, and /bin/busybox
	Read system binary files	Reads the binary of the currently running process: /proc/self/exe
	Read system and process information	Reads files /proc/mounts, /proc/1/ for mount and process info
	List GPU information	Uses nvidia-smi –list-gpus to list GPU data, filtering for unique names
Trace Cleaning	Create backdoor	Establishes a reverse shell or downloads malicious code for remote control
	Modify BusyBox permissions	Copies and modifies BusyBox permissions for unrestricted future use
	Disable and clear shell history	Disables and clears shell history to prevent future command logging
	Clear log files	Uses "head -c 0 >filename" to delete file contents
Persistent access	Create new user, modify user group attributes	Creates a new user and adds it to the sudo group
	Change key file attributes	Uses chattr and lockr to change .ssh directory attributes
	Modify user and group permissions	Modifies /etc/sudoers to change permissions
	Delete all deny access rules	Empties /etc/hosts.deny to remove access restrictions
	Create backdoor	Establishes a reverse shell or downloads malicious code for remote control
	Terminate processes	Searches and terminates processes not using standard libraries or specific markers
	Terminate specific processes	Searches and terminates processes with specific names

TABLE IV
HIGH-FREQUENCY ATTACK COMMANDS

Rank	Commands in 2021 and 2022	Commands in 2024
1	uname	cd ∼ ; chattr -ia .ssh ; lockr -ia .ssh
2	echo -e "{user}\n{passwd}" |passwd |bash	cd && rm -rf .ssh && mkdir .ssh && echo {ssh_rsa_key} >>.ssh/authorized_keys && chmod -R ∼ /.ssh && cd ∼
3	echo "{user}\n{passwd}" |passwd	uname
4	cat /proc/cpuinfo |grep name |wc -l	echo {user}:{passwd} |chpasswd |bash
5	cat /proc/cpuinfo |grep name |head -n 1 |awk '{ print $4, $5, $6, $7, $8, $9 ; }'	shell
6	which	sh
7	free -m |grep Mem |awk '{ print $2, $4, $5, $6, $7 }'	enable
8	w	system
9	crontab	cat /proc/cpuinfo |grep name |wc -l
10	ls -lh $(which ls)	cat /proc/cpuinfo |grep model |grep name |wc -l
11	cat /proc/cpuinfo |grep model |grep name |wc -l	df -h |head -n {number} |awk '{ print $2 ; }'
12	top	w
13	lscpu |grep Model	which
14	cd && rm -rf .ssh && mkdir .ssh && echo {ssh_rsa_key} >>.ssh/authorized_keys && chmod -R ∼ /.ssh && cd ∼	whoami
15	echo {user}:{passwd} |chpasswd |bash	ls -lh $(which ls)
16	sh	rm -rf {file1} ; rm -rf {file2} ; pkill -9 {Process1} ; pkill -9 {Process2} ; echo >{file3} ; pkill -9 {Process4} ;
17	shell	cat /proc/cpuinfo |grep name |head -n 1 |awk '{ print $4,$5,$6,$7,$8,$9 ; }'
18	enable	crontab
19	echo	free -m |grep Mem |awk '{ print $2, $4, $5, $6, $7 }'
20	system	lscpu |grep Model

data gathering and are actively seeking to ensure they can maintain control and possibly avoid remediation efforts. The increased focus on persistence tactics indicates an evolution in the attackers' objectives, where establishing and maintaining a strong foothold within the compromised system has become the primary goal.

Persistence involves techniques that adversaries employ to maintain access to systems despite restarts and other disruptions that could cut off their connection. Examples of these strategies include modifying user accounts and creating SSH private keys, which have been widely used from 2021 to 2024. Notably, these procedures have become more frequently utilized and increasingly diverse in 2024. During 2021-2022, persistence commands ranked 2nd, 3rd, 14th, and 15th, whereas in 2024, these commands soared to the 1st, 2nd, and 4th positions. Moreover, the usage of commands like "cd ∼; chattr -ia .ssh; lockr -ia .ssh" in 2024 further demonstrates that attackers are actively expanding their procedures to enhance their persistence mechanisms.

Defense evasion consists of techniques that adversaries use to avoid detection throughout their compromise. These techniques include uninstalling or disabling security software, as well as obfuscating or encrypting data and scripts. Notably, there were no defense evasion commands among the top 20 commands in 2021-2022. However, by 2024, defense evasion commands became more prevalent. For example, the command "rm -rf {file1}; rm -rf {file2}; pkill -9 {Process1}; pkill -9 {Process2}; echo >{file3}; pkill -9 {Process4};" emerged as the 16th most common command. This command is used to delete sensitive files and terminate system defense processes, illustrating a strategic shift toward more aggressive evasion tactics.

The comparative analysis of high-frequency attack commands between 2021-2022 and 2024 reveals a significant shift in the execution tactic employed by adversaries. In 2021-2022, shell commands such as ”shell” (rank 17), ”sh” (rank 16), ”enable” (rank 18), ”echo” (rank 19), and ”system” (rank 20) were less prevalent, suggesting that adversaries relied on other techniques for executing malicious code on compromised systems. However, in 2024, these commands have soared in popularity, with ”shell” (rank 5), ”sh” (rank 6), ”enable” (rank 7), and ”system” (rank 8) now ranking among the top 10 most frequently used commands. This dramatic increase in the use of shell commands for execution indicates a shift in adversary tactics, where they are now more likely to leverage command-line interfaces and built-in system tools to run malicious code. This change in tactics may be attributed to the increasing effectiveness of security controls and detection mechanisms that focus on other execution methods, such as malware or exploit-based attacks. As a result, adversaries are turning to more basic, yet still effective, techniques to evade detection and maintain control over compromised systems. The rise in the use of shell commands for execution underscores the importance of monitoring and securing command-line interfaces, as well as implementing robust log analysis and anomaly detection mechanisms to identify and respond to these threats in a timely manner.

In summary, the comparative analysis of high-frequency attack commands between 2021-2022 and 2024 reveals significant shifts in adversary tactics across multiple stages of the attack lifecycle. Attackers have increasingly employed complex commands, prioritized persistence tactics over passive discovery, and adopted more aggressive defense evasion techniques. Furthermore, the dramatic increase in the use of shell commands for execution indicates a shift toward leveraging command-line interfaces and built-in system tools to run malicious code.

2) System-Type-Specific Commands Adoption: We investigate the system-specific commands used by attackers, targeting Ubuntu-simulated honeypots during the periods 2021-2022 and 2024. Ubuntu, a widely used Linux distribution, includes a set of standard commands for operations such as file management, system information retrieval, networking, user management, shell scripting, and package management.

Our analysis shows that in the period 2021-2022, approximately 41.43% of the commands used by attackers corresponded with standard Ubuntu commands. By 2024, this proportion had increased to approximately 48.44%. The observed increase suggests a growing proficiency among attackers in leveraging system-specific commands when targeting Ubuntu systems. This trend indicates an enhanced understanding and adaptation to the specific environments of their targets, potentially leading to more effective and sophisticated attacks.

These findings underscore the critical need for continuous monitoring and adaptation of security measures. As attackers increasingly exploit system-specific knowledge, defensive strategies must evolve to mitigate the risks associated with these advanced tactics.

TABLE V
ATTACK SESSION OVERVIEW

Characteristics	2021-2022	2024
Session counts	5,363	6,658
Session Kinds	4161	3118
Session pattern	144	245
Number of Session Patterns	145	156
Average Session Length	12.65	5.00
Average Length per Session Kind	10.97	11.80

C. Session Quantitative Analysis

We conduct an in-depth analysis of the data collected by the honeypot from the session dimension to gain deeper insights into the attack landscape. First, we categorize the vast amount of session information based on session patterns, which are collections of command patterns. Sessions with the same pattern are considered to belong to the same attack entity. we analyze the length of sessions to gauge the persistence and engagement level of attackers, revealing the changes in the attack landscape.

Table V presents an overview of the attack sessions observed during the periods of 2021-2022 and 2024. The data reveals a notable increase in the number of session kinds from 5,363 in 2021-2022 to 6,659 in 2024, indicating a growing diversity in attacker behaviors. However, the number of distinct session patterns only increases slightly from 145 to 156 over the same period, suggesting that attackers may employ similar tactics across different sessions.

The significant decrease in average session length from 12.65 commands per session in 2021-2022 to just 5.00 commands per session in 2024 can be attributed to two primary factors. First, attackers have become more efficient in their tactics, adopting more targeted approaches that allow them to achieve their objectives with fewer commands. Second, attackers have enhanced their reconnaissance capabilities, enabling them to quickly assess the suitability of a target environment and distinguish genuine systems from honeypots or decoys. These factors have resulted in shortened session lengths, as attackers can make rapid decisions and execute their attacks more effectively. Consequently, this increased efficiency and heightened sensitivity pose significant challenges for defenders, as it narrows the window for attack detection and response, necessitating the implementation of more proactive and adaptive security measures to effectively counter these evolving threats.

Despite the decrease in average session length, the average length per session kind remains relatively stable, with 10.97 commands per session kind in 2021-2022 and 11.80 commands per session kind in 2024. This consistency suggests that while the overall number of sessions has increased, the complexity of individual session kinds has not changed significantly.

TABLE VI
TACTICS USAGE FREQUENCIES COMPARISON

Tactic	Technology	2021-2022	2024
Discovery	T1082, T1083, T1087, T1424, T1033, T1016, T1518, T1069, T1124, T1614	33.13%	18.05% ↓
Collection	T1005	0.02%	0.21% ↑
Command and Control	T1105, T1071	1.60%	1.79% ↑
Credential Access	T1003	0.00%	0.01% ↑
Defense Evasion	T1222, T1070, T1655, T1562, T1027, T1564, T1548	7.58%	38.77% ↑
Execution	T1059	18.01%	14.88% ↓
Impact	T1489, T1496	0.02%	0.07% ↑
Persistence	T1098, T1053, T1547	19.75%	13.11% ↓
Privilege Escalation	T1098, T1053, T1547, T1548	19.88%	13.12% ↓

By classifying attacker sessions based on their command patterns, we gain valuable insights into the evolving tactics and strategies employed by attackers. This information can be used to develop more effective detection and mitigation techniques, ultimately enhancing the security posture of shell services and protecting against emerging threats.

D. Evolution of ATT&CK technology and tactic

We compare attacker strategies observed between 2021–2022 and 2024, focusing on how adversarial behaviors evolve under the MITRE ATT&CK Matrix for Enterprise. As shown in Tables I andV, the timespan and variety of shell attack sessions in the 2021–2022 and 2024 datasets are relatively balanced, allowing a meaningful comparative analysis. We select the ATT&CK Matrix for its comprehensive representation of post-compromise tactics, closely mirroring real-world shell service attack patterns.

Notably, each command within a shell session is treated as the smallest unit of analysis, and all relevant techniques invoked by that command are identified. We then merge these command-level techniques into a single sequence per session, ensuring that multiple commands employing the same technique are consolidated into one occurrence.

By systematically examining each phase of the attack chain from initial discovery to impact, we highlight notable shifts in the tactical progression of shell-based intrusions. Our comparison centers on two main aspects: (1) changes in the usage frequency of techniques across different tactics, and (2) variations in the integration of ATT&CK techniques during shell sessions.

1) Usage Frequency of ATT&CK Technology and Tactic: Tables VI and VII list how the usage frequencies of ATT&CK tactics and technologies shifted between 2021–2022 and 2024. In the following, we analyze key trends in tactics such as Defense Evasion, Discovery, and Credential Access, as well as their implications for adversarial strategies.

TABLE VII
TECHNIQUES USAGE FREQUENCIES COMPARISON

Technology	2021,2022	2024
T1016 System Network Configuration Discovery	0.085%	1.751% ↑
T1033 System Owner/User Discovery	0.004%	1.589% ↑
T1069 Permission Groups Discovery	0%	0.016% ↑
T1082 System Information Discovery	18.294%	6.894% ↓
T1083 File and Directory Discovery	8.058%	4.499% ↓
T1087 Account Discovery	11.551%	4.382% ↓
T1124 System Time Discovery	0.004%	0.002% ↓
T1424 Process Discovery	5.726%	1.835% ↓
T1518 Software Discovery	0.004%	0.133% ↑
T1614 System Location Discovery	0%	0.006% ↑
T1005 Data from Local System	0.023%	0.240% ↑
T1071 Application Layer Protocol	0%	0.019% ↑
T1105 Ingress Tool Transfer	2.113%	2.070% ↓
T1003 OS Credential Dumpin	0%	0.017% ↑
T1059 Command and Scripting Interpreter	18.043%	15.815%↓
T1053 Scheduled Task/Job	5.731%	1.583% ↓
T1098 Account Manipulation	20.268%	13.740%↓
T1547 Boot or Logon Autostart Execution	0.069%	0.005% ↓
T1548 Abuse Elevation Control Mechanism	0.173%	0.019% ↓
T1027 Obfuscated Files or Information	0.499%	1.582% ↑
T1070 Indicator Removal	1.834%	17.176%↑
T1222 File and Directory Permissions Modification	5.840%	22.948%↑
T1562 Impair Defenses	0%	1.585% ↑
T1564 Hide Artifacts	0.005%	0.146% ↑
T1655 Masquerading	1.650%	1.874% ↑
T1489 Service Stop	0.021%	0.073% ↑
T1496 Resource Hijacking	0.005%	0.004% ↓

Defense Evasion - heightened preference and increased diversity: In 2024, attackers shifted from rapid intrusion tactics to more prolonged, covert strategies, resulting in a pronounced rise in Defense Evasion as a primary approach. Notably, the employment of T1070 Indicator Removal jumped from 1.83% to 17.18%, while T1222 File and Directory Permissions Modification increased from 5.84% to 22.95%. Moreover, a new technique, T1562 Impair Defenses, appeared during this period. These trends suggest that adversaries placed greater emphasis on evading detection, concealing their activities, and undermining defensive systems to remain inconspicuous and minimize detection risk.

Discovery - decrease in overall usage but broader technique diversity and more balanced distribution: Although the overall usage frequency of the Discovery Tactic dropped from 33.13% (2021–2022) to 18.05% (2024), the diversity of associated techniques expanded markedly. New additions include T1069 Permission Groups Discovery and T1614 System Location Discovery, which uncover information regarding user-group permissions and geographic location, respectively. A few Discovery techniques experienced a particularly sharp increase in 2024, such as T1033 System Owner/User Discovery (0.004% ->1.589%) and T1518 Software Discovery (0.004% ->0.133%), suggesting that adversaries have shifted toward faster methods for gathering user and software information. Furthermore, the standard deviation of Discovery technique usage declined from 0.0645 to 0.0239, indicating a more balanced distribution of Discovery methods overall.

Persistence, Privilege Escalation, and Execution - a noticeable decline in traditional attack strategies: Amid adversaries' growing emphasis on stealthiness and concealment, conventional tactics such as Persistence, Privilege Escalation, and Execution have seen a decrease in usage. Notably,

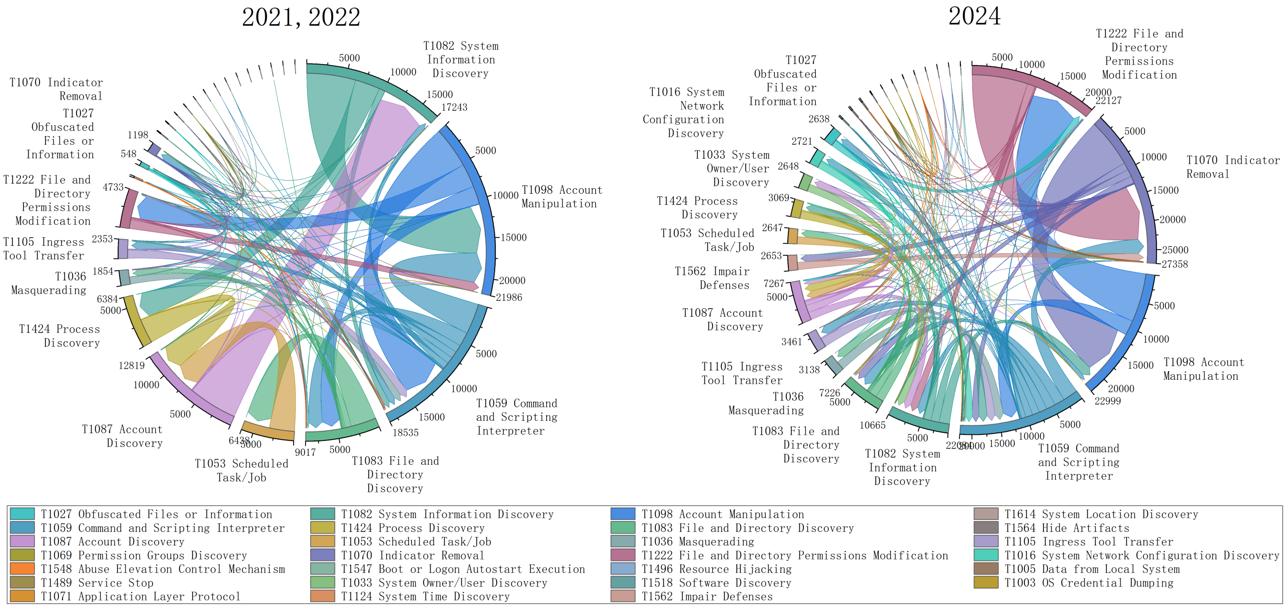

Fig. 1. Intrinsic Relationships Among ATT&CK Technique Sequences

Persistence and Privilege Escalation methods show a significant overlap, primarily relying on T1053 Scheduled Task/Job, T1098 Account Manipulation, and T1547 Boot or Logon Autostart Execution. Meanwhile, Execution heavily depends on T1059 Command and Scripting Interpreter and T1053 Scheduled Task/Job. This trend underscores the attackers' shift away from overt compromise methods in favor of more discreet strategies aimed at evading detection.

Credential Access - newly emergent in 2024: Credential Access first appeared as a distinct tactic in 2024 shell attack dataset and is represented by a single technique: T1003 OS Credential Dumping. This technique is designed to extract user credentials from a system, complementing the findings from 2021-2022, which only included T1098 Account Manipulation, a technique associated with shell account login operations (e.g., modifying or creating system accounts). Note that the latter technique is categorized under the Persistence and Privilege Escalation tactics.

Collection & Command and Control & Impact - a slight increase in previously low-usage tactics: While attackers in 2024 predominantly focus on stealthiness, tactics such as Collection, Command and Control, and Impact have shown modest yet noteworthy growth. All three maintained usage rates below 2% during 2021–2022. In the Collection category, T1005 Data from Local System shows an uptick from 0.023% in 2022 to 0.240% in 2024, illustrating a heightened interest in extracting local data for further exploitation. Command and Control also experienced a minor rise, highlighted by the advent of T1071 Application Layer Protocol at 0.019%. Meanwhile, T1489 Service Stop under the Impact tactic increased from 0.021% to 0.073%. These developments suggest that attackers are balancing an emphasis on stealthiness with a measured but clear intent to exert direct influence over system integrity and resources, signifying a more multifaceted strategy that extends beyond just concealment.

2) Intrinsic Relationships Among ATT&CK Technique Sequences: Figure 1 features two chord diagrams, illustrating the ATT&CK techniques leveraged during 2021–2022 and 2024, respectively, along with their interlinked relationships. These visual representations underscore the dynamic interconnections among techniques, highlighting both the prevalence of individual methods and their interplay within complex attack vectors.

Compared to 2021–2022, the number of ATT&CK techniques increased from 22 to 27 in 2024, reflecting a diversification of techniques employed by attackers. Additionally, the usage of techniques in 2024 became more evenly distributed, with smaller gaps in frequency among different techniques, indicating a more balanced application of methods. Furthermore, the connections between techniques in the 2024 chord diagram appear denser, suggesting a higher level of interdependence and coordination among techniques, a clear indication of the increasingly sophisticated and integrated attack strategies.

To further quantify these visual trends, Table VIII compares the top ten ordered 2-tuples of techniques observed in 2020–2021 and 2024, and it indicates each ordered pair's percentage share of all observed two-technique combinations. Several salient patterns emerge from this comparison, which are detailed as follows.

Shift to anti-forensics and stealth: In 2020–2021, the highest-frequency pairs revolve around reconnaissance and execution activities like "T1082 System Information Discovery → T1098 Account Manipulation". By contrast, 2024's top ordered 2-tuples emphasize permission modifications and log removal—" T1222 File & Directory Permissions Modification → T1070 Indicator Removal".

Decline of overt execution: The Command & Scripting Interpreter (T1059) drops from appearing in three of the top five transitions in 2020–2021 to much lower-ranked positions

TABLE VIII

COMPARISON OF TOP 10 MITRE ATT&CK TECHNIQUE TUPLES: 2020–2021 VS. 2024

Rank	2020–2021	2024
1	T1082 System Information Discovery → T1098 Account Manipulation (12.016%)	T1222 File and Directory Permissions Modification → T1070 Indicator Removal (14.024%)
2	T1087 Account Discovery → T1082 System Information Discovery (12.006%)	T1098 Account Manipulation → T1222 File and Directory Permissions Modification (12.288%)
3	T1098 Account Manipulation → T1059 Command and Scripting Interpreter (10.159%)	T1070 Indicator Removal → T1098 Account Manipulation (12.286%)
4	T1059 Command and Scripting Interpreter → T1098 Account Manipulation (7.523%)	T1059 Command and Scripting → T1070 Indicator Removal (3.784%)
5	T1053 Scheduled Task/Job → T1087 Account Discovery (6.216%)	T1098 Account Manipulation → T1059 Command and Scripting Interpreter (3.706%)
6	T1083 File and Directory Discovery → T1053 Scheduled Task/Job (6.214%)	T1082 System Information Discovery → T1098 Account Manipulation (2.614%)
7	T1082 System Information Discovery → T1424 Process Discovery (6.115%)	T1083 File and Directory Discovery → T1083 File and Directory Discovery (2.564%)
8	T1424 Process Discovery → T1087 Account Discovery (5.994%)	T1059 Command and Scripting Interpreter → T1105 Ingress Tool Transfer (2.278%)
9	T1098 Account Manipulation → T1222 File and Directory Permissions Modification (5.806%)	T1036 Masquerading → T1059 Command and Scripting Interpreter (2.184%)
10	T1098 Account Manipulation → T1083 File and Directory Discovery (4.418%)	T1083 File and Directory Discovery → T1036 Masquerading (2.125%)

in 2024. Crucially, it is no longer predominantly paired with the obvious Account Manipulation technique (T1098); instead, T1059 co-occurs with a more diverse set of stealthy tactics, including Defense Evasion (T1070, T1036), and Command and Control through Ingress Tool Transfer (T1105). This shift underscores adversaries' movement away from a single, conspicuous execution step toward more covert, multifaceted operational patterns.

Emergence of staging and masquerading: New high-ranking entries such as Indicator Removal (T1070) and Masquerading (T1036) reflect growing emphasis on covert payload delivery and disguise within the attack pipeline.

These observations quantify and reinforce the chord-diagram insights: adversaries have transitioned from relatively linear and noisy "reconnaissance → execution → persistence" flows toward highly integrated, stealth-first pipelines that prioritize anti-forensics and permission manipulation as core enablers of complex attack campaigns.

E. Summary of Attack Situation Changes

A comparative analysis of the 2021–2022 and 2024 attack datasets from Cowrie reveals that shell-based intrusions have undergone major transformations over a span of time. These changes include a higher diversity and complexity of individual commands, stronger emphasis on sustaining long-term access and avoiding detection, briefer yet more targeted infiltration attempts, broader distribution of attack techniques, and deeper integration of multiple methods into cohesive attack chains. The following points elaborate on these key shifts:

1) **Expanded Command Repertoires and Complexity:** Attackers now employ a broader range of shell commands and exhibit novel behaviors to achieve objectives such as stealthiness, information gathering, and maintaining access. This evolution not only underscores a rise in tactical flexibility but also complicates detection efforts.
2) **Shift Toward Defense Evasion:** Traditional attack tactics such as discovery, execution, persistence, and privilege escalation appear less dominant than before, giving way to concerted efforts for sustained long-term access, stealthiness, and defense evasion.
3) **Shorter and More Targeted Sessions:** As command complexity increases, attackers increasingly consolidate multiple shell commands into a single execution to maximize efficiency. Furthermore, adversaries have become adept at quickly discerning high-value targets from honeypots and real servers through minimal interaction, posing a challenge to the effectiveness of current honeypot deception techniques.
4) **Growing Technique Diversity and Balanced Distribution:** Under the MITRE ATT&CK framework, the emergence of new tactics such as credential access and an overall increase in techniques variety illustrates the adversaries' heightened adaptability. Furthermore, the frequency gap between different attack techniques is shrinking, indicating a more balanced deployment of attack methods in contemporary intrusion scenarios.
5) **More Cohesive Utilization of Techniques:** Throughout the attack process, multiple ATT&CK techniques are increasingly integrated in denser, more tightly knit sequences. This intensified chaining diminishes the effectiveness of single-point detection and calls for more comprehensive and coordinated defensive efforts.

V. DEFENSIVE RECOMMENDATIONS

The evolving shell attack landscape necessitates updated defensive strategies to address emerging threats. Based on our findings, we propose four key recommendations to counter emerging threats.

1. Adopting AI-Driven Command Analysis for Complex Shell Attacks: As shell attack commands grow in complexity, traditional rule-based detection methods struggle to adapt to new and obfuscated techniques employed by attackers. In contrast, by leveraging machine learning models trained on high-quality datasets, AI-driven command analysis offers superior adaptability and analytical capabilities, making it more suitable against increasingly complex shell-based attacks.

2. Strengthening Defenses Against Defense Evasion Techniques: The growing emphasis on defense evasion tactics—such as modifying file permissions, deleting system logs, tampering with firewall rules, stealing sensitive files, and employing file obfuscation—demands enhanced protective measures. System administrators should implement stricter access control policies, ensuring that critical functions (e.g., logging, access permissions, and sensitive data operations) are rigorously monitored and protected. Additionally, deploying behavioral analysis tools to detect unusual activities, such as sudden log deletions or unauthorized permission changes, can help mitigate evasion attempts.

3. Enhancing Honeypot Capabilities to Counter Advanced Attackers: Our analysis reveals a trend toward shorter interactions with honeypots, suggesting that attackers are becoming more adept at identifying and avoiding decoy systems. This may be due to the rigid and easily identifiable features of traditional honeypots, which fail to convincingly mimic high-value targets. To address this problem, next-generation honeypots should be designed to emulate high-configuration hosts and provide dynamic, attack-tailored responses. By increasing the realism and flexibility of honeypots, defenders can better engage and study advanced adversaries, gaining deeper insights into their attack tactics and techniques.

4. Expanding Detection Coverage for Diverse Attack Techniques: The diversification and balanced distribution of attack techniques underscore the need for comprehensive detection strategies. Defenders should broaden the scope of their detection to cover a wider range of MITRE ATT&CK techniques, including less common but increasingly prevalent methods. Regularly updating detection rules and incorporating threat intelligence feeds can help ensure that defenses remain aligned with the evolving threat landscape. Additionally, conducting red team exercises to simulate emerging attack scenarios will enable organizations to test and refine their defensive capabilities, ensuring preparedness against new threats.

VI. DISCUSSION

The primary efforts of this work fall into two areas: constructing a comprehensive shell-based request-response dataset and conducting an in-depth analysis of evolving attack behaviors. Both rely on data gathered from Cowrie honeypots spanning 2021–2022 and 2024. Based on the detailed examination of the collected data, we further establish the dataset's richness, representativeness, and novelty. To create a standardized request-response dataset, we develop a multidimensional, in-depth, and verifiable framework for evaluating honeypot systems, while simultaneously offering crucial data support for AI-driven honeypot research. We compare datasets from different periods to identify emerging attack vectors, investigate how attack patterns have evolved at both command and session levels, and utilize the MITRE ATT&CK framework to assess overall trends. The results indicate significant developments in shell-based intrusions, including more extensive and complex command usage, heightened focus on defense evasion, shorter yet more targeted attack processes, greater technique diversity with a more balanced distribution, and deeper integration of multiple methods. These findings illuminate the rapidly evolving nature of threats and underscore the need for more robust and adaptive defensive strategies.

Despite the comprehensive nature of our collected datasets, some limitations remain. Honeypot data inherently reflects only a subset of attack activities, and factors such as geographic deployment and the presence of cloud or IoT environments may skew the view of global attack behaviors. In our session-level analyses, an increasing number of attackers maintained only minimal interactions with honeypots, implying improved recognition of decoy systems. This underscores the need to enhance honeypot deception capability for ensuring deeper and more sustained attacker engagement. Although our approach illuminates adversarial practices against single shell services, it cannot fully capture real-time coordination among multiple attack vectors or complex tactics that surpass the scope of a single honeypot setup.

Moving forward, the relationship between high-quality data analysis and optimization of honeypot performance will be more tightly integrated, with each domain driving and reinforcing advancements in the other. Strengthening honeypot designs to increase their deceptive capabilities and capture more advanced attack behaviors will yield higher-quality data and provide deeper insights into adversarial intent and development. Moreover, high-quality honeypot data can facilitate significant progress in AI and machine learning research, driving innovations in detection, prevention, and response mechanisms and ultimately fortifying cybersecurity defenses in an ever-changing threat landscape.

VII. CONCLUSION

This study introduced and examined a comprehensive shell-based request-response dataset derived from Cowrie honeypots across two distinct time frames (2021–2022 and 2024). By systematically comparing adversarial commands andsession behaviors, and leveraging the MITRE ATT&CK framework, we highlighted major shifts in intruder techniques, including heightened command complexity, shift toward defense evasion, shorter yet more strategically orchestrated attacks, and broadened use of diverse approaches. More importantly, the standardized dataset developed here can serve as a benchmark for evaluating honeypot systems and provide an essential resource for AI-driven cybersecurity research. In demonstrating how quickly attacker methods adapt, our findings underscore the growing importance of dynamic and data-rich honeypot implementations in fortifying defenses against evolving security threats.

References

[1] Alexander, V., and Richard, C. Bitter harvest: Systematically fingerprinting low- and medium-interaction honeypots at internet scale. In *12th USENIX Workshop on Offensive Technologies* (2018).

[2] Anagnostakis, K. G., Sidiroglou, S., Akritidis, P., Xinidis, K., Markatos, E., and Keromytis, A. D. Detecting targeted attacks using shadow honeypots. In *14th USENIX Security Symposium* (2005).

[3] Başer, M., Güven, E. Y., and Aydin, M. A. Ssh and telnet protocols attack analysis using honeypot technique: Analysis of ssh and telnet honeypot. In *2021 6th International Conference on Computer Science and Engineering* (2021).

[4] Hofstede, R., Hendriks, L., Sperotto, A., and Pras, A. Ssh compromise detection using netflow/ipfix. *ACM SIGCOMM computer communication review 44*, 5 (2014), 20–26.

[5] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685* (2021).

[6] Huang, L., and Zhu, Q. Duplicity games for deception design with an application to insider threat mitigation. *IEEE Transactions on Information Forensics and Security* (2021).

[7] Javed, M., and Paxson, V. Detecting stealthy, distributed ssh bruteforcing. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security* (2013), pp. 85–96.

[8] Jiangyi Deng, Xinfeng Li, Y. C. Y. B. H. W. Y. L. T. W. X. Raconteur: A knowledgeable, insightful, and portable llm-powered shell command explainer. In *NDSS* (2025).

[9] Kim, J., Marin, E., Conti, M., and Shin, S. Equalnet: A secure and practical defense for long-term network topology obfuscation. In *Proceedings of the 29st Annual Network and Distributed System Security Symposium* (2022), Association for Computing Machinery.

[10] Ling, Z., Luo, J., Xu, D., Yang, M., and Fu, X. Novel and practical sdn-based traceback technique for malicious traffic over anonymous networks. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications* (2019), pp. 1180–1188.

[11] Liu, H., Zhou, Y., Fang, B., Sun, Y., Hu, N., and Tian, Z. Phcg: Plc honeypoint communication generator for industrial iot. *IEEE Transactions on Mobile Computing* (2024).

[12] López-Morales, E., Rubio-Medrano, C., Doupé, A., Shoshitaishvili, Y., Wang, R., Bao, T., and Ahn, G.-J. Honeyplc: A next-generation honeypot for industrial control systems. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security* (2020).

[13] Niels, P. A virtual honeypot framework. In *13th USENIX Security Symposium* (2004).

[14] OOSTERHOF, M. Cowrie. [Online]. https://github.com/micheloosterhof/cowrie.

[15] Piet, J., Sharma, A., Paxson, V., and Wagner, D. Network detection of interactive SSH impostors using deep learning. In *32nd USENIX Security Symposium* (Anaheim, CA, 2023), pp. 4283–4300.

[16] Priya, V. D., and Chakkaravarthy, S. S. Containerized cloud-based honeypot deception for tracking attackers. *Scientific Reports* (2023).

[17] Ring, M., Wunderlich, S., Scheuring, D., Landes, D., and Hotho, A. A survey of network-based intrusion detection data sets. *Computers & security 86* (2019), 147–167.

[18] Rrushi, J. L. Dnic architectural developments for 0-knowledge detection of opc malware. *IEEE Transactions on Dependable and Secure Computing* (2018).

[19] Sasaki, T., Fujita, A., Gañán, C. H., van Eeten, M., Yoshioka, K., and Matsumoto, T. Exposed infrastructures: Discovery, attacks and remediation of insecure ics remote management devices. In *2022 IEEE Symposium on Security and Privacy (SP)* (2022), pp. 2379–2396.

[20] Shea, R., and Liu, J. Understanding the impact of denial of service attacks on virtual machines. In *2012 IEEE 20th International Workshop on Quality of Service* (2012), pp. 1–9.

[21] Sladić, M., Valeros, V., Catania, C., and Garcia, S. Llm in the shell: Generative honeypots. In *2024 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)* (2024), pp. 430–435.

[22] Strom, B. E., Applebaum, A., Miller, D. P., Nickels, K. C., Pennington, A. G., and Thomas, C. B. Mitre att&ck: Design and philosophy. In *Technical report*. The MITRE Corporation, 2018.

[23] Tian, K., Mitchell, E., Yao, H., Manning, C. D., and Finn, C. Fine-tuning language models for factuality, 2023.

[24] Torabi, S., Bou-Harb, E., Assi, C., Karbab, E. B., Boukhtouta, A., and Debbabi, M. Inferring and investigating iot-generated scanning campaigns targeting a large network telescope. *IEEE Transactions on Dependable and Secure Computing* (2020).

[25] Wahab, O. A., Bentahar, J., Otrok, H., and Mourad, A. Resource-aware detection and defense system against multi-type attacks in the cloud: Repeated bayesian stackelberg game. *IEEE Transactions on Dependable and Secure Computing* (2019).

[26] Wang, Y., Su, Z., Benslimane, A., Xu, Q., Dai, M., and Li, R. Collaborative honeypot defense in uav networks: A learning-based game approach. *IEEE Transactions on Information Forensics and Security* (2023).

[27] Wang, Z., Yan, Y., Yan, Y., Chen, H., and Yang, Z. {CamShield}: Securing smart cameras through physical replication and isolation. In *31st USENIX Security Symposium* (2022).

[28] Wang, Z., You, J., Wang, H., Yuan, T., Lv, S., Wang, Y., and Sun, L. Honeygpt: Breaking the trilemma in terminal honeypots with large language model. *arXiv preprint arXiv:2406.01882* (2024).

[29] Yang, Z., Liu, X., Li, T., Wu, D., Wang, J., Zhao, Y., and Han, H. A systematic literature review of methods and datasets for anomaly-based network intrusion detection. *Computers & Security 116* (2022), 102675.

[30] Zhan, Z., Xu, M., and Xu, S. Characterizing honeypot-captured cyber attacks: Statistical framework and case study. *IEEE Transactions on Information Forensics and Security* (2013).