Information Security Project Report

Mohammad Yehya K213309, Student, FAST NUCES
Daniyal Haider K213433, Student, FAST NUCES
Hasan Iqbal K213297, Student, FAST NUCES

***Abstract*—**

A. Good Bot, Bad Bot: Characterizing Automated Browsing Activity

The growth of the web has increased the prevalence of both legitimate and malicious bots. Legitimate bots enhance search engines, while malicious bots exploit vulnerabilities and conduct brute-force attacks. This paper introduces Aristaeus, a system designed for honeysite deployment and analysis to detect bot behavior. Through a seven-month global study of 100 honeysites, Aristaeus identified more than 76,396 malicious bots among 287,017 unique IPs. It revealed significant impersonation by bots, using falsified TLS and HTTP headers. The findings contribute to understanding bot traffic and improving detection tools. ***Index Terms*—Bots, Honeysites, Fingerprinting, Cybersecurity, Web Exploits, TLS**

B. Black Widow: Blackbox Data-driven Web Scanning

Modern web applications are central to our digital lives but remain vulnerable due to their complexity and reliance on third-party components. Blackbox scanning, which relies on no prior knowledge about application behavior, faces challenges in crawling and vulnerability detection due to dynamic content and inter-page dependencies. This report introduces Black Widow, a novel approach that addresses these challenges by combining navigation modeling, traversing workflows, and tracking inter-state dependencies. Black Widow demonstrates substantial improvements in code coverage and vulnerability detection, including discovering vulnerabilities missed by existing tools.
***Index Terms*—Web Application Security, Blackbox Scanning, Dynamic Web Pages, Vulnerability Detection, Cross-Site Scripting (XSS), Taint Tracking, Navigation Modeling**

C. Detecting AI Trojans Using Meta Neural Analysis

Trojan attacks on neural networks pose significant security threats, allowing adversaries to embed malicious functionalities into models while maintaining their performance on benign inputs. This paper introduces a novel framework, Meta Neural Trojan Detection (MNTD), to detect such attacks without assumptions on the attack strategies or requiring white-box access to the models. MNTD uses a meta-classifier trained on shadow models to distinguish Trojaned from benign models based on a carefully optimized query set. Experimental results across diverse datasets and attack types demonstrate MNTD's superior detection performance, achieving up to 97% detection AUC and robustness against adaptive attacks. The approach shows generalizability to unforeseen attacks, highlighting its potential in securing AI systems.
***Index Terms*—Trojan attacks, Neural networks, Backdoor Detection, Meta-classifier, Shadow models, Jumbo learning, Query optimization, Black-box Access, Adaptive attacks, Detection AUC, Generalizability, Security-critical applications, Machine learning security, Trigger patterns, Poisoning attacks**

D. Self-Supervised Euphemism Detection and Identification for Content Moderation

Fringe groups increasingly use euphemisms—ordinary words with secret meanings—to bypass content moderation on social media. Traditional keyword-based filters fail to handle the nuanced contexts of such terms. This paper proposes unsupervised algorithms leveraging masked language models and self-supervised learning for euphemism detection and identification. The algorithms achieve significantly higher detection accuracies (30–400%) over prior methods and represent the first known approach to identifying euphemisms' precise meanings. Our findings advance automated content moderation, making strides against policy evasion by exploiting linguistic context effectively.
***Index Terms*—Adversarial Content, Keyword Evasion, Social Media Platforms, Euphemism Detection, Content Moderation**

E. REVEALER: Detecting and Exploiting Regular Expression Denial-of-Service Vulnerabilities

Regular Expression Denial-of-Service (ReDoS) is a critical performance vulnerability affecting modern web applications reliant on regex patterns. The vulnerabilities stem from extended features in regex engines, leading to super-linear matching complexities under specific attack strings. This study introduces REVEALER, a novel hybrid system that combines static and dynamic analysis to identify and exploit ReDoS vulnerabilities efficiently. REVEALER statically locates vulnerable structures within regex patterns and dynamically validates them by generating attack strings that trigger recursive backtracking. An evaluation of REVEALER on 29,088 regex patterns shows its superiority over state-of-the-art tools, identifying 213 previously unknown vulnerabilities and outperforming competitors by 140.64%. Furthermore, REVEALER uncovered 45 vulnerabilities in popular open-source applications, underscoring its effectiveness and applicability.
***Index Terms*—Regex Matching Algorithms, Super-Linear Time Complexity, Vulnerabilities, Extended Regexes,REVEALER, Static Analysis, Dynamic Analysis, Vulnerable Patterns, Attack String Generation**

F. Using Selective Memoization to Defeat Regular Expression Denial of Service (ReDoS)

Regular expressions (regexes) are critical tools in modern computing but pose significant risks as vectors for Regular Expression Denial of Service (ReDoS) attacks. Such attacks exploit the super-linear time complexity of many regex engines, potentially disrupting services. This research introduces selective memoization schemes and an encoding strategy to eliminate ReDoS vulnerabilities effectively. Evaluated on real-world datasets, these approaches reduce the space costs of memoization by orders of magnitude and ensure linear worst-case time complexity for the majority of regexes, offering a practical, backwards-compatible defense for legacy systems.
***Index Terms*—ReDoS (Regular Expression Denial-of-Service),Vulnerability detection,Regex (Regular Expressions),**

Extended regex features, Catastrophic backtracking, Hybrid analysis (static and dynamic), e-NFA (Extended Nondeterministic Finite Automaton), Attack string generation, ReDoS mitigation, Performance evaluation

I. Introduction

A. *Good Bot, Bad Bot: Characterizing Automated Browsing Activity*

Web bots play a dual role, aiding both legitimate services like search engines and malicious actors conducting cyber-attacks. Industry reports indicate bots constitute 37.2percent of web traffic, with 64.7percent of this being malicious. Existing bot-detection techniques face challenges due to spoofed identities and limited datasets for training. This paper presents Aristaeus, a scalable honeysite system that traps bots to gather behavioral and technical data. Aristaeus enables the study of malicious bots through real deployments, augmented with fingerprinting and logging.

B. *Black Widow: Blackbox Data-driven Web Scanning*

Ensuring the security of modern web applications is crucial due to their widespread usage and complexity. Vulnerabilities such as Cross-Site Scripting (XSS) remain a significant threat, often bypassing traditional mitigation techniques. Existing blackbox scanning tools struggle with dynamic behaviors, asynchronous processes, and inter-page dependencies. Black Widow overcomes these issues by creating a dynamic, data-driven approach that captures inter-state dependencies, navigates complex workflows, and tracks control and data flows.

C. *Detecting AI Trojans Using Meta Neural Analysis*

The rapid adoption of deep neural networks (DNNs) in critical applications such as autonomous driving and cybersecurity has exposed them to Trojan attacks. These attacks inject malicious functionalities, triggered by specific patterns, without affecting the model's performance on benign data. Existing detection methods are often limited by their reliance on assumptions about the attack or their need for white-box access. Addressing these limitations, this study proposes MNTD, a black-box detection framework leveraging meta-neural analysis. MNTD generalizes across datasets and tasks, identifying Trojaned models by training a meta-classifier on shadow models with diverse attack settings. This paper evaluates MNTD against state-of-the-art detection methods and demonstrates its efficacy in diverse scenarios, including adaptive attacks.

D. *Self-Supervised Euphemism Detection and Identification for Content Moderation*

Social media platforms struggle with content moderation as users employ euphemisms to evade keyword-based bans. Manual moderation is unsustainable, costly, and psychologically taxing. Euphemism detection and interpretation remain under-explored in natural language processing (NLP). This research introduces a two-step pipeline: detection of euphemisms using self-supervised learning and identification of their meanings via coarse-to-fine-grained classification. The proposed methods aim to assist human moderators by automating the discovery and interpretation of euphemisms in textual data.

E. *REVEALER: Detecting and Exploiting Regular Expression Denial-of-Service Vulnerabilities*

Regular expressions (regexes) are foundational tools in programming, facilitating pattern matching for diverse applications. Modern regex engines extend their capabilities by incorporating advanced features such as backreferences and named groups. However, these extensions increase complexity, exposing applications to ReDoS attacks. This study introduces REVEALER, a hybrid system that integrates static modeling and dynamic verification to detect and exploit ReDoS vulnerabilities effectively. By bridging gaps in existing approaches, REVEALER addresses both the structural complexity and runtime performance issues associated with modern regex engines.

F. *Using Selective Memoization to Defeat Regular Expression Denial of Service (ReDoS)*

Regexes are indispensable for pattern matching in software development. However, their widespread adoption exposes critical vulnerabilities, notably the ReDoS attack. These attacks exploit super-linear time complexities inherent in backtracking regex engines, threatening service availability. Despite the severity of the issue, existing defenses—such as regex optimization, regex replacement, or engine overhaul—face challenges in compatibility, soundness, and cost. This study proposes a selective memoization strategy that guarantees linear time complexity while maintaining compatibility with legacy systems.

II. Literature Review

A. *Good Bot, Bad Bot: Characterizing Automated Browsing Activity*

1) Bot Detection Techniques: Traditional detection methods rely on browser and behavioral fingerprinting. However, the difficulty in distinguishing bots from users arises due to dynamic bot evolution, identity spoofing, and CAPTCHA bypass mechanisms.

2) Honeysites: Honeysites mimic real websites but are specifically designed to attract bots. By isolating bot traffic, honeysites avoid the complexities of mixed user-bot environments, enabling a focused study on malicious behavior.

3) Existing Challenges:

- Dependence on reliable datasets
- Bot operators leveraging residential networks for evasion
- The increasing power of bots in spoofing identities

B. *Black Widow: Blackbox Data-driven Web Scanning*

Blackbox scanners face several challenges when handling modern web applications:

1) Navigation Modeling:: Dynamic web pages with JavaScript events, forms, and iframes require robust modeling of server-side and client-side interactions.
Existing approaches fail to capture inter-state dependencies, such as relationships between user actions on one page and changes in another.

2) Traversing Workflows:: Complex workflows (e.g., multi-step forms) involve sequential dependencies that are difficult to navigate.
Traditional methods fail to manage authentication, session handling, and context-sensitive data.

3) Inter-State Dependencies:: Vulnerabilities often depend on subtle relationships between inputs and application states. Identifying these requires dynamic tracking of taint sources (input fields) and sinks (output locations).

C. *Detecting AI Trojans Using Meta Neural Analysis*

Previous works on Trojan detection can be categorized into model-level, input-level, and dataset-level approaches. Model-level methods like Neural Cleanse and DeepInspect detect anomalies in the model's behavior but rely on restrictive assumptions. Input-level approaches, such as STRIP and SentiNet, analyze model responses to specific inputs, while dataset-level techniques focus on identifying poisoned training data. However, these methods often fail to generalize to diverse attack strategies or require significant model access. The proposed MNTD addresses these gaps by offering a generalizable, black-box detection framework.

D. *Self-Supervised Euphemism Detection and Identification for Content Moderation*

Research in euphemism detection spans supervised, semi-supervised, and unsupervised approaches. Supervised models rely heavily on annotated datasets, while semi-supervised approaches combine labeled and unlabeled data but suffer from domain-specific limitations. Context-free embeddings (e.g., word2vec) dominate unsupervised learning but lack context-awareness, limiting their efficacy. More recent advancements, such as BERT-based embeddings, enable context-sensitive learning but remain underutilized in euphemism detection. Self-supervised methods, like masked language models, offer promising solutions by leveraging inherent data patterns to perform unsupervised learning tasks.

E. *REVEALER: Detecting and Exploiting Regular Expression Denial-of-Service Vulnerabilities*

Regex vulnerabilities are well-documented, with empirical studies identifying super-linear behavior in up to 10% of regexes. Previous defenses include adopting linear-time regex engines like RE2, applying piecemeal regex refactoring, and introducing resource caps. However, these approaches often compromise functionality or introduce significant overhead. Memoization, a technique proven effective in parsing contexts, has not been widely applied due to its prohibitive space costs. This research builds on the theoretical foundation of memoization, adapting it for practical regex defenses against ReDoS.

F. *Using Selective Memoization to Defeat Regular Expression Denial of Service (ReDoS)*

Research on ReDoS vulnerabilities highlights two primary detection approaches: static analysis and dynamic fuzzing. While static methods rely on identifying vulnerable patterns in regex structures, they often yield false positives and struggle with extended features. Dynamic methods, such as fuzzing, generate malicious inputs but are computationally intensive and less effective for complex patterns. Hybrid approaches, though promising, have not fully addressed the challenges posed by context-sensitive grammar inherent in extended regexes. This study builds on these methodologies, proposing a comprehensive system that models vulnerabilities in e-NFA structures and automates attack string generation with higher precision.

III. System Design

A. *Good Bot, Bad Bot: Characterizing Automated Browsing Activity*

Aristaeus comprises three main components:

1) Honeysites: A honeysite is a real deployment of a web application, augmented with different fingerprinting techniques, and increased logging. Hosting vulnerable web applications (e.g., WordPress, Joomla) with fingerprinting modules.

2) Log Aggregation: It consists of: Log Correlation, Querying Panel, and Session Generation. It's a centralized collection of data across honeysites.

3) Analysis Engine: The finding suggests that most bot requests come from either infected residential devices, or using residential devices as a proxy to evade IP-based blocklists. It correlates bot behaviors, IP addresses, and requests for detailed insights.

B. *Black Widow: Blackbox Data-driven Web Scanning*

Black Widow addresses these challenges through three pillars:

1) Navigation Modeling: A graph-based representation of application states and transitions.
Includes server-side dependencies and client-side interactions like JavaScript events and form submissions.

2) Workflow Traversing: A traversal algorithm to execute workflows safely and efficiently.
Uses a heuristic to prioritize "safe" edges, such as GET requests, and reconstructs workflows as needed.

3) Tracking Inter-State Dependencies: Implements dynamic taint tracking to link input fields (sources) with their reflections in outputs (sinks).
Injects unique tokens into input fields and monitors for their reappearance in application responses.

C. *Detecting AI Trojans Using Meta Neural Analysis*

MNTD employs a three-step pipeline:

1) *Shadow Model Generation:* A diverse set of benign and Trojaned shadow models is created using jumbo learning, which samples attack settings from a general distribution.

2) *Meta-Training:* A meta-classifier is trained on the shadow models using optimized queries to distinguish Trojaned from benign models. Query tuning ensures the representation vectors maximize discriminatory features.
3) *Target Model Detection:* The meta-classifier evaluates the target model based on its responses to the optimized queries, predicting whether it is Trojaned.

D. *Self-Supervised Euphemism Detection and Identification for Content Moderation*

The proposed pipeline consists of two main components:

1) *Euphemism Detection:* A masked language model (MLM) predicts euphemistic terms based on their sentence-level contexts. A filtering mechanism eliminates generic, non-informative contexts.
2) *Euphemism Identification*: Using coarse-to-fine-grained classifiers, this module maps detected euphemisms to specific target meanings. A binary classifier first narrows down relevant contexts, followed by a multi-class classifier that predicts precise meanings.

Datasets from domains like drugs, weapons, and sexuality provide the textual corpus for validation.

E. *REVEALER: Detecting and Exploiting Regular Expression Denial-of-Service Vulnerabilities*

REVEALER is designed to analyze regexes in two phases:

1) *Static Analysis:* Regexes are parsed into a simplified e-NFA structure (E-TREE), enabling efficient traversal and identification of vulnerable patterns, such as "Loop in Loop" or "Branch in Loop."
2) *Dynamic Analysis:* Attack strings are crafted by simulating regex matching, leveraging structural insights to generate cores that cause catastrophic backtracking. The system iteratively validates potential vulnerabilities by observing runtime performance.

F. *Using Selective Memoization to Defeat Regular Expression Denial of Service (ReDoS)*

he proposed system employs selective memoization, focusing on vertices in the regex's non-deterministic finite automaton (NFA) where ambiguity compounds. Three schemes are introduced:

1) *Full Memoization (Qall):* Tracks all simulation positions for guaranteed unambiguous behavior.
2) *Selective Memoization (Qin-deg¿1):* Targets vertices with high in-degree to mitigate convergence-related redundancy.
3) *Cycle Ancestors (Qancestor):* Focuses on vertices related to cycles, ensuring bounded ambiguity.

IV. Implementation

A. *Good Bot, Bad Bot: Characterizing Automated Browsing Activity*

1) Honeysite Deployment: 100 honeysites were deployed using vulnerable templates across three continents. To ensure clean datasets, domains were registered with no prior history.

2) Bot Fingerprinting: Aristaeus applies three fingerprinting techniques:

- **Browser Fingerprinting:** Captures JavaScript capabilities and unique features like canvas rendering.
- **Behavioral Analysis:** Monitors bot interactions, including violations of robots.txt directives.
- **TLS Fingerprinting:** Detects discrepancies between stated user-agent and TLS handshake.

B. *Black Widow: Blackbox Data-driven Web Scanning*

Black Widow integrates its techniques into a scanner built on a modern browser using Selenium and a state-of-the-art JavaScript engine. Key components include:

1) Dynamic Analysis:: Hooks into JavaScript functions and DOM modifications to monitor behavior.

2) Infinite Crawl Avoidance:: Implements heuristics to detect and prevent endless loops, such as calendar navigation.

3) XSS Detection:: Dynamically verifies injected payloads to minimize false positives.

C. *Detecting AI Trojans Using Meta Neural Analysis*

The framework was implemented across various datasets, including MNIST, CIFAR10, and speech and text datasets. Trojaned models were generated using modification and blending attacks, while benign models were trained on clean data. A small subset of the clean data was used for training the shadow models. MNTD's meta-classifier was a two-layer neural network optimized with gradient-based methods, and the query set was iteratively tuned for maximum performance.

D. *Self-Supervised Euphemism Detection and Identification for Content Moderation*

Detection:

- MLM extracts masked sentences and ranks potential euphemism replacements.
- Informative sentences are filtered using a threshold based on the rank of target-related terms.

Identification:

- A binary classifier filters sentences unrelated to target categories.
- A fine-grained classifier identifies the euphemism's target keyword using a labeled dataset constructed via self-supervised masking.

Both components were implemented using BERT for contextual embeddings and evaluated on diverse datasets.

E. *REVEALER: Detecting and Exploiting Regular Expression Denial-of-Service Vulnerabilities*

The prototype implementation of REVEALER focuses on the Java regex engine. It statically identifies vulnerabilities using the E-TREE representation and dynamically simulates regex matching to generate attack strings. A validation mechanism ensures only true vulnerabilities are reported. The system employs thresholds for attack string length and matching steps to refine analysis, balancing precision with computational efficiency.

F. Using Selective Memoization to Defeat Regular Expression Denial of Service (ReDoS)

A prototype was developed within a backtracking regex engine, incorporating selective memoization and RLE-based memo function representation. The system was tested on a large corpus of real-world regexes, including both linear and super-linear variants. Additional support for extended regex features, such as zero-width assertions and backreferences, was implemented to evaluate the broader applicability of the approach.

V. Results and Analysis

A. Good Bot, Bad Bot: Characterizing Automated Browsing Activity

Over seven months, Aristaeus captured:

- **26.4 million requests** from 287,017 unique IPs.
- **57 percent malicious traffic**, including brute-force attacks and exploitation attempts.

1) Geographical Trends: Most requests originated from the U.S., China, and Brazil, with residential networks accounting for 64.37

2) Malicious Bot Behavior:

- **Brute Force Attack** Targeting login pages, with WordPress being the most attacked platform.
- **Fingerprinting Attempts** ots probed for application vulnerabilities using requests like /CHANGELOG.txt.

3) Spoofing Analysis: **86.2 percent of bots impersonated browsers**, primarily Chrome and Firefox, while using simple HTTP libraries like wget or curl.

B. Black Widow: Blackbox Data-driven Web Scanning

1) Code Coverage: Black Widow was tested on ten web applications, including legacy platforms (e.g., SCARF, phpBB) and modern production software (e.g., WordPress, PrestaShop). It improved code coverage by 63–280 compared to other scanners, with unique lines of code discovered in applications like PrestaShop and Vanilla.

2) Vulnerability Detection: The scanner found 25 unique vulnerabilities, including six previously unknown XSS vulnerabilities in production applications like WordPress, osCommerce, and PrestaShop. Unlike other tools, Black Widow produced no false positives, thanks to its dynamic verification mechanism.

3) Improved Coverage:: Black Widow consistently outperformed other scanners in discovering new code paths and parameters, particularly in dynamic and complex applications.

4) Better Vulnerability Detection:: By analyzing inter-state dependencies, Black Widow identified multi-step vulnerabilities that traditional tools missed, such as stored XSS in multi-page workflows.

5) No False Positives:: The use of dynamic payload verification significantly reduced errors in vulnerability reports

C. Detecting AI Trojans Using Meta Neural Analysis

MNTD achieved exceptional detection performance, with an average AUC of 97% across datasets and attack types. It outperformed existing methods in generalizability and robustness. Against adaptive attacks, MNTD demonstrated a detection AUC of 90%, showcasing its resilience even when attackers had system knowledge. The system also generalized well to unforeseen attack strategies, such as latent and parameter attacks, and unforeseen model architectures.

D. Self-Supervised Euphemism Detection and Identification for Content Moderation

Detection:
The proposed approach outperformed baselines like Word2Vec and prior models such as CantReader, achieving precision improvements of up to 400%. It demonstrated robust performance across drugs, weapons, and sexuality datasets.

Identification:
For mapping euphemisms to specific keywords, the algorithm surpassed random baselines and clustering-based methods, achieving top-1 accuracies of 20–33% across datasets. Coarse-to-fine-grained classification significantly enhanced performance compared to single-stage classifiers.

Analysis:
The results revealed new euphemisms absent in the ground truth, showcasing the model's ability to generalize. Limitations included challenges in adversarial settings and variations in dataset specificity.

E. REVEALER: Detecting and Exploiting Regular Expression Denial-of-Service Vulnerabilities

REVEALER was evaluated against a benchmark of 29,088 regexes, detecting 450 vulnerabilities with zero false positives. It identified 213 previously unknown vulnerabilities, outperforming state-of-the-art tools like ReScue (187 vulnerabilities), RXXR2 (112 vulnerabilities), and Rexploiter (63 vulnerabilities). Additionally, REVEALER demonstrated robustness in handling extended features and complex structures, with significant improvements in detection rates for "Loop in Loop" and "Loop after Loop" patterns. The average processing time per regex was 0.0076 seconds, highlighting its computational efficiency.

F. Using Selective Memoization to Defeat Regular Expression Denial of Service (ReDoS)

1) *Time Complexity*: The evaluation confirmed linear time complexity for all tested super-linear regexes under the proposed schemes. Case studies, including high-profile examples like Stack Overflow's outage regex, demonstrated significant reductions in processing time.
2) *Space Complexity*: Selective memoization dramatically reduced space costs compared to full memoization. The RLE representation achieved constant space for most regexes, with significant compression observed for repetitive patterns.

3) *Extended Features:* Memoization for extended regex features, such as backreferences, achieved substantial improvements. However, backreferences still exhibited higher space-time tradeoffs due to their context-sensitive nature.

VI. Conclusion

A. *Good Bot, Bad Bot: Characterizing Automated Browsing Activity*

The study demonstrates the effectiveness of honeysites in isolating and analyzing bot behavior. By identifying malicious patterns and TLS inconsistencies, Aristaeus provides actionable insights for improving bot-detection systems.

B. *Black Widow: Blackbox Data-driven Web Scanning*

Black Widow demonstrates the potential of data-driven approaches for blackbox web scanning. By combining navigation modeling, workflow traversal, and dependency tracking, it provides significant improvements in code coverage and vulnerability detection. The tool's effectiveness in finding previously undiscovered vulnerabilities makes it a valuable asset for securing modern web applications.

C. *Detecting AI Trojans Using Meta Neural Analysis*

MNTD represents a significant advancement in Trojan detection, providing a robust and generalizable solution for securing AI systems. Its ability to function in a black-box setting without assumptions about attack strategies makes it suitable for practical deployment in diverse applications. Future work will explore enhancing MNTD's scalability and extending its applicability to other machine learning paradigms.

D. *Self-Supervised Euphemism Detection and Identification for Content Moderation*

The study introduces state-of-the-art methods for euphemism detection and identification, addressing critical gaps in automated content moderation. By leveraging context-aware embeddings and self-supervised learning, the proposed framework reduces reliance on manually curated datasets and achieves robust detection and mapping. Future work could extend these methods to multimedia content and adversarial scenarios, further enhancing their practical utility.

E. *REVEALER: Detecting and Exploiting Regular Expression Denial-of-Service Vulnerabilities*

This study presents REVEALER, a hybrid approach for detecting and exploiting ReDoS vulnerabilities in extended regexes. By integrating static and dynamic analyses, REVEALER achieves superior accuracy and efficiency compared to existing tools. Its ability to handle complex regex features and provide actionable insights positions it as a valuable tool for enhancing application security. Future work includes expanding feature support and optimizing the dynamic analysis process to address remaining limitations.

F. *Using Selective Memoization to Defeat Regular Expression Denial of Service (ReDoS)*

This study presents a robust, practical solution to ReDoS vulnerabilities through selective memoization and efficient encoding strategies. By ensuring linear worst-case time complexity and minimizing space costs, the proposed approach provides a sound, backwards-compatible defense. Future work will focus on refining these techniques for broader adoption and extending their applicability to more complex regex features.

References

[1] Li, X., et al. "Good Bot, Bad Bot: Characterizing Automated Browsing Activity." 2021 IEEE Symposium on Security and Privacy (SP).

[2] A. Shirokova, "Cms brute force attacks are still a threat." [Online]. Available: https://blogs.cisco.com/security/cms-brute-force-attacks-arestill-a-threat

[3] T. Canavan, CMS Security Handbook: The Comprehensive Guide for WordPress, Joomla, Drupal, and Plone. John Wiley and Sons, 2011.

[4] A. G. Lourenc¸o and O. O. Belo, "Catching web crawlers in the act, "in Proceedings of the 6th international Conference on Web Engineering, 2006, pp. 265–272

[5] S. Sivakorn, J. Polakis, and A. D. Keromytis, "I'm not a human: Breaking the google recaptcha," Black Hat, 2016.

[6] D. Canali and D. Balzarotti, "Behind the Scenes of Online Attacks: an Analysis of Exploitation Behaviors on the Web," in Proceedings of the 20th Network and Distributed System Security Symposium (NDSS), 2013.

[7] Google, "Vulnerability Reward Program: 2019 Year in Review,"https://security.googleblog.com/2020/01/vulnerabilityreward-program-2019-year.html, 2020

[8] S. Innovation, "Google Awards 1.2 Million in Bounties Just for XSS Bugs,"https://blog.securityinnovation.com/googleawards-1.2-million-in-bounties-just-for-xss-bugs, 2016.

[9] Bugcrowd, "The State of Crowdsourced Security 2019,"https://www.bugcrowd.com/, 2020.

[10] InfoSecurity, "XSS is Most Rewarding Bug Bounty as CSRF is Revived news/xss-bug-bounty-csrf-1-1-1-1/, 2019.

[11] A. Riancho, "w3af - open source web application security scanner," 2007.

[12] OWASP, "Owasp zed attack proxy (zap)," 2020.

[13] X. Xu, Q. Wang, H. Li, N. Borisov, C. A. Gunter, and B. Li, "Detecting AI Trojans Using Meta Neural Analysis," arXiv preprint, vol. arXiv:1910.03137v4, 2020.

[14] W. Zhu, H. Gong, R. Bansal, Z. Weinberg, N. Christin, G. Fanti, and S. Bhat, "Self-supervised euphemism detection and identification for content moderation," arXiv preprint arXiv:2103.16808, Mar. 2021

[15] Liu, Y., Zhang, M., & Meng, W. (2021). Revealer: Detecting and Exploiting Regular Expression Denial-of-Service Vulnerabilities. IEEE Symposium on Security and Privacy.

[16] Shen, X., et al. (2020). ReScue: A Fuzzer for Detecting ReDoS Vulnerabilities.

[17] Ekdahl, M. (2019). RXXR2: Improved Static Analysis for ReDoS Detection.

[18] Davis, J. C., et al. (2021). Using Selective Memoization to Defeat Regular Expression Denial of Service (ReDoS). IEEE Symposium on Security and Privacy.

[19] Thompson, K. (1968). Programming Techniques: Regular Expression Search Algorithm. Communications of the ACM.

[20] Crosby, S. A., & Wallach, D. S. (2003). Denial of Service via Algorithmic Complexity Attacks. USENIX Security Symposium.