JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

A MULTI-FACETED APPROACH TO CYBERSECURITY: MISCONFIGURATION ANALYSIS, WEB SERVR FINGERPRINTING, AND IoT THREAT DETECTION

Aafrin Julaya, Dr. Akshara Dave
Department of Computer Science, Department of Computer Science
Indus University, Ahmedabad 382115,India

ABSTRACT- This study delves into significant cybersecurity issues, including the detection of malicious hosts, the security of authentication systems, the impact of misconfigurations, and advanced fingerprinting techniques. It introduces a sophisticated fingerprinting framework designed to identify unwanted network traffic and analyzes how security misconfigurations can lead to extensive breaches, especially within IPv6 networks. Authentication databases are a common target for attackers due to inadequate security practices. In response, this research proposes a robust password security framework that employs salting, hashing, and iteration strategies. The investigation also highlights the risks of privilege escalation in cloud-based identity management platforms, such as Azure Active Directory, and offers strategies for mitigation. Moreover, the research showcases innovative fingerprinting approaches, including a two-stage method for identifying Tor hidden services, HTTPFuzz for web server identification via HTTP request fuzzing, and Dossier, a tool that tracks provenance to analyze configuration-based attack vectors. These contributions play a vital role in strengthening security by tackling critical vulnerabilities across various digital environments.

Keywords: Cyber security Misconfigurations, Fingerprinting Techniques, Authentication Security

I. INTRODUCTION

This study addresses the pressing challenge of cybersecurity within legacy database systems, emphasizing the vulnerabilities linked to outdated authentication methods. As the pace of digital transformation quickens, the interconnectedness of systems increases the risk of exposing sensitive information to various cyber threats. Many older infrastructures do not comply with current security standards, rendering them vulnerable to credential theft and unauthorized access. We propose an innovative strategy to enhance authentication security in legacy databases by tackling issues such as inadequate password storage, vulnerability to SQL injection attacks, and obsolete encryption methods. The proposed framework incorporates cryptographic best practices like salting, hashing, and iterative processing, which not only bolster security but also maintain system performance and user-friendliness by [5] through comprehensive testing and security evaluations, this research assesses the proposed solution's effectiveness in addressing real-world vulnerabilities. The results provide practical guidance for securing authentication databases, contributing to ongoing efforts to reinforce cybersecurity while remaining compatible with existing legacy systems.

II. Global Network Threat Assessment: Major Insights

The architecture of the internet connects billions of devices, creating a vast ecosystem where malicious actors often operate. By combining the AR framework with the rDSN module, distributed sensor networks can work together to improve the detection of harmful network activity. While Network Telescope captures SYN packets across all ports, Honeypot sensors only record attacks aimed at their specific listening services, leading to a significantly lower number of detected incidents compared to the broader monitoring capabilities of Network Telescope.
Data from RUScope sensors indicates that the main geographic sources of hostile network traffic are: China - 16% United States - 10% Russian Federation - 7% Research conducted by Kim et al. (2012) previously validated that China and the United States are major sources of global cybersecurity threats. Service Port Vulnerability Evaluation The following analysis outlines the most commonly targeted service ports and their corresponding percentages of malicious host activity:

II.I Commonly Targeted Network Ports in Cyber Attacks

Cybercriminals frequently focus on certain service ports to take advantage of weaknesses and obtain unauthorized access. The ports listed below have been recognized as the most commonly attacked:

Port	Service Type	Proportion of Malicious Hosts
443	Encrypted Web Services (HTTPS)	29.96%
21	File Transfer Services (FTP)	16.84%
143	Email Access Protocol (IMAP)	4.53%
88	Common Web Services (Unsecured HTTP)	4.31%

Table I. Most Targated Service Ports

These **management service ports** (e.g., RDP, FTP, SSH) constitute half of the top 10 list and account for **88.01% of open ports** across all malicious hosts. The prevalence of **open PPTP (1723/tcp) ports** is particularly concerning, as compromised hosts could allow attackers access to internal networks via VPN entry points.

III. Real-Time Detection using the rDSN Module

The rDSN module, implemented on three data sensors, facilitates the near real-time recognition of harmful hosts. This module offers an extensive overview of malicious activities from various distributed sources. This research seeks to:

- Describe hosts that generate malicious traffic on the Internet.
- Examine demographic trends related to malicious hosts.
- Establish host re-identification strategies for dynamic IP environments.
- Enhance techniques for identifying and monitoring malicious activities.

IV. Strengthening Legacy Databases: A Contemporary Approach to Identity Verification

Authentication databases are common cybercrime targets due to outdated security, despite available best practices from groups like OWASP and IEEE. To prevent password theft, the article recommends using strong hashing methods (e.g., SHA-256/512), adding a 32-byte salt, and rehashing at least 1,000 times, following standards like the OWASP Password Storage Cheat Sheet and RFC 2898.

The Payment Card Industry Data Security Standard (PCIDSS) created by PCI-SSC requires that organizations meet at least the above criteria to be considered compliant [1] However, despite the availability of a wealth of information on best practice implementation, many organizations have yet to implement best practices and compliance due to budget constraints, time factors, and reluctance to replace existing infrastructure identify by the [2] Often, even organizations that do attempt to implement protective measures fail to implement best practices correctly. Subsequent investigations of password breaches have revealed that several companies maintain substandard implementations [3] Compromised databases from Yahoo, Friend Finder, and Hello Kitty found that passwords were stored in plain text or hashed using outdated one-way encryption functions such as MD5 and SHA-128. Additionally, all organizations refrained from using salts and repetitions in their password protection mechanisms [4] The organizations involved were not following basic best practice recommendations and therefore were not in compliance.

VI. Mathematical Frameworks in Cyber security

VI. I. Cryptographic Hashing Theorems

One-Way Function: A key principle is the one-way function, which allows for easy calculation of the hash (h(x) from an input x) while making it computationally challenging to determine the original input (x) from the hash value (h(x)).

Theorem	Particulars	Real-Word Example
Deterministic	Identical inputs will consistently yield the same hash.	The **RockYou breach** showed users pick weak passwords with low entropy, making them vulnerable to attacks.
Collision Resistance	It should be extremely difficult to identify two distinct inputs that generate the same hash output.	**Google & CWI Amsterdam (2017)** demonstrated an SHA-1 collision, leading to SHA-256 adoption.
Pre-image Resistance	It should be extremely challenging to discover an input that results in a specified hash output.	The **2016 LinkedIn breach** exposed weak SHA-1 hashed passwords, making preimage attacks feasible.

Table II. Description of hashing theorems

VI.II. Probability & Statistics in Intrusion Detection

Theorem	Particulars	Real-World Example
Bayes' Theorem	P(X)=Occurrence of X/Total Events	Estimate likelihood of attack
Bayes Theorem	P(A|B)= P(B|A)*P(A)/P(B)	Malware detection tools had a **5% FPR**, incorrectly flagging 5 out of 100 legitimate apps. It aids in identifying emails as either spam or legitimate by analyzing the presence of particular words, as discussed by upGrad. In the realm of fraud detection, it assesses the probability of a transaction being fraudulent by utilizing historical data and risk factors.
Hypothesis Testing	H0: No attack, H1: Attack	Decision making using p-value, z-score

Table III. Various theorems for probability & statistics in intrusion detection

VI.III. Graph Theory in Network Security

A *Graph* is a mathematical structure used to model pairwise relations between objects.
Mathematically:
A graph G=(V,E)
Where:

- V= Set of nodes/vertices (e.g., computers, routers)
- E = Set of edges (connections between nodes)

Let's develop a basic proof of concept demonstrating how graph theory can improve security: Assertion: Analyzing graphs can minimize the likelihood of unnoticed intrusions by improving the positioning of IDS.
Proof (Conceptual):

1. Represent the network as a graph
2. Let $C \subseteq V$ be the set of **compromised nodes.**
3. The objective is to position the least number of Intrusion Detection Systems (IDS) to monitor all traffic to and from compromised nodes.
4. The Vertex Cover Problem:

- Finding the smallest set $S \subseteq V$, such that for every edge $(u,v) \in E$, at least one of u or v is in S.
- This directly applies to IDS placement: placing IDS at vertices in S ensures all data flowing through any edge is monitored.

5. Given that vertex cover is an established challenge in graph theory that has approximation algorithms, we can effectively implement IDS by using these algorithms [15].
6. This minimizes the attack surface and guarantees optimal surveillance, thereby demonstrating that optimization through graph theory improves network security.

Domain	**Graph Theory use case**
Enterprise Networks	Identifying attack paths through the use of attack graphs
IoT Security	Detecting anomalies by analyzing changes in communication graphs
Blockchain & P2P	Managing trust through the use of weighted trust graphs
Malware Propagation	Representing malware spread as a problem of graph traversal
Cloud Network Security	Determining the best placement of firewall rules using flow graphs

Table IV. Graph theory use case

IV. Game Theory in Cyber security

Game theory offers a strategic approach for examining and modeling the conflicts between assailants and defenders in the realm of cyber security. It is utilized to predict the actions of attackers, develop the best defensive strategies, and allocate resources efficiently.

Illustration of a Game Model:

Let's analyze a basic game involving an attacker and a defender:
Participants: Attacker (A), Defender (D)
Choices: A: {target server 1, target server 2} D: {protect server 1, protect server 2}
Outcomes:
If D safeguards the server being attacked → A receives -1, D receives +1
If D protects the alternative server → A gets +1, D receives -1

This is a zero-sum game, and the Nash equilibrium entails both players selecting their strategies randomly.

V. Complexity Theory in Cryptographic Algorithms

Complexity theory categorizes computational problems according to the resources (time, space) required for their resolution.

- P (Polynomial Time): Problems that can be solved efficiently.
- NP (Nondeterministic Polynomial Time): Problems for which solutions can be verified efficiently.
- NP-Hard / NP-Complete: The most challenging problems within NP.
- EXP: Problems that can be solved in exponential time.

Algorithm	**Hard Problem**	**Complexity**	**Security Insight**
RSA	Integer Factorization	No known algorithm that operates in polynomial time (probably exponential)	Secure as the challenge of factoring large numbers is significant
ECC (Elliptic Curve Cryptography)	Elliptic Curve Discrete Logarithm Problem	Sub-exponential	Offers robust security with smaller key sizes due to the complexity of its mathematics
AES (Symmetric Encryption)	Brute-force Key Search	Exponential ($O(2^n)$)	Highly secure when using large key sizes (e.g., 256-bit)

Table V. Various complexity algorithms

VI. Collaborative execution of password policies

A review of frequently used password security protocols was performed to assess their effectiveness in improving the security of user accounts [15] Enforcing policies that require specific password length, complexity, and expiration decreases the threat of unauthorized access to personal accounts via brute force and dictionary attacks [8]. Nevertheless, passwords saved in authentication databases continue to be susceptible to exposure through injection attacks.

Password Enhancement Tools

The functionality of several password management applications, such as Password Agent, Lucent Personalized Web Assistant (LPWA), PwdHash, and Password Multiplier, has been tested. The research developed a secure authentication system for legacy databases using Apache's mod_rewrite, enhancing password protection with salting, hashing, and iteration [5] While testing confirmed its reliability and compatibility, vulnerabilities like SQL injection risks from dynamic queries were identified, prompting recommendations for stronger encryption and prepared statements.

VII. A Practical Assessment of Misconfigurations in Internet Services and Their Security Consequences.

POTENTIAL OUTCOMES (THEORETICAL INSTANCE)

SERVICE	FREQUENT CONFIGURATION ERRORS	% OF IMPACTED SYSTEMS
Apache Web Server	Directory listing allowed	24%
MySQL	Remote root access permitted	11%
Amazon S3	Read/write access publicly available	31%
Redis	No password protection enabled	27%

Table VI. An approach of misconfiguration internet services

VIII. Analysis of Fingerprinting Techniques for Tor Hidden Services and Website Fingerprinting Attacks

Research Problem: Website fingerprinting attacks aim to scrutinize encrypted network traffic by extracting multiple characteristics:
X={x1,x2,...,xn}
These characteristics may encompass packet size, packet order, and packet direction, enabling attackers to deduce which hidden services users are visiting on the Tor network.

Proposed Approach: A Two-Phase Fingerprinting Framework

Phase 1: Detection of Hidden Services

This phase's objective is to ascertain whether a specific traffic trace
f1(T)={1,0} if T corresponds to a hidden service otherwise
T belongs to a Tor hidden service or not.

Phase 2**:** Identification of Specific Hidden Services

After being identified as hidden service traffic, the subsequent step is to pinpoint the exact hidden service.
S={S1,S2,...,Sm}
f2(T)=Si, Si ∈ S

Performance Analysis & Scalability:

A large-scale and realistic dataset D was developed specifically for evaluating fingerprinting attacks against hidden services.

- *Phase 1:* The detection model

F_1(T)= (T) exhibited high accuracy and practical applicability.

- *Phase 2:* Current classification techniques face major challenges when scaling to large numbers of hidden services. Specifically, as the number of services mmm increases:

$$\lim_{m\to\infty} \text{Accuracy}(f_2) \longrightarrow 0$$

Accuracy tends to decline significantly.

Final Remarks: This research underscores that while contemporary classifiers are proficient at identifying hidden services; their capacity to effectively differentiate among a vast array of individual hidden services is constrained. The findings highlight the scalability challenges that emerge when implementing fingerprinting attacks in practical Tor scenarios.

XI. Detecting hidden services

Our fingerprinting technique consists of two classification phases. In phase 1, we determine whether a client has connected to a HS(hidden services) or visited a regular web page. Furthermore, we also want to answer the question of whether it is possible to establish a new, i.e., communication with a HS. h. We use the FPs(fingerprinting) of known HSs to identify previously unseen HSs. Once a HS connection is detected, in phase 2 we try to pinpoint exactly which HS content was visited [6] Below, we describe and evaluate both classification phases in detail.

XI. Exploiting Misconfiguration Vulnerabilities in Microsoft's Azure Active Directory for Privilege Escalation Attacks

Microsoft's cloud services, such as Azure Active Directory (AAD), are becoming more important, especially for managing identities in cloud setups. However, there are security issues that cloud systems encounter, mainly because of weaknesses and mistakes in identity management systems [[9]

- This research looks into and assesses vulnerable misconfigurations in AAD that could be exploited for attacks that escalate privileges.

- Two situations are studied: changing group settings and the Managed Identity function on virtual devices.
- Tests show that unauthorized access to important data can be achieved.
- The research suggests ways to reduce these risks, like separating important systems, and stresses the importance of more research in this area.

X. Dossier: A Tool for Tracing Provenance in Cybersecurity

In the realm of cyber security, it is essential to comprehend the source, movement, and timeline of data or actions within a system to detect attacks, track adversaries, and maintain system integrity. Dossier is a provenance tracing tool specifically developed to address these requirements in contemporary cyber security landscapes.

Problem:

Traditional causal analysis lacks application-level insights, particularly regarding **configuration changes**, leading to **inaccurate attack attribution** and **insufficient tracebacks**.

Proposed Solution:

Dossier, Dossier, a tool for tracing provenance, strengthens the Linux testing framework by monitoring configuration modifications, which are a significant attack vector. It offers a comprehensive overview of event histories and causal links to enhance forensic analysis.

Core Features:

- **File Configuration Monitoring** – Observes changes to sensitive files through a kernel module.
- **Memory Configuration Monitoring** – Employs LLVM instrumentation to record changes in configuration made dynamically.
- **In-Depth Lineage Graph** – Combines system and configuration logs for thorough reconstruction of attacks.

Dossier serves as a formidable resource in the cybersecurity toolkit, allowing organizations to progress past basic detection towards a comprehensive understanding and tracking of security events. Through the use of provenance tracing, Dossier aids security analysts in uncovering more profound insights into attacks, enhancing response times, and bolstering overall system robustness.

XII. HTTPFuzz: Web Server Fingerprinting with HTTP Request Fuzzing

Web server based fingerprinting is a type of fingerprinting that allows security researchers, penetration testers, and attackers to distinguish servers based on the information exposed by the server. A common method to hide this information is to use fingerprint mitigation techniques. In this work, we present a novel approach to fingerprint web server software, independent of the fingerprinting mitigation techniques applied [10] The premise of our approach is based on the simple recognition that web servers handle different types of HTTP requests differently. They use fuzzing techniques to intelligently and adaptively select HTTP requests that can trick the server into revealing service level information [11]

Network Analysis & Fingerprinting Tools

Tool	Function	Key Features
Nmap	Network analysis & fingerprinting	Uses TCP probes & regex matching; inefficient for large-scale scans.
HTTPrint	HTTP server identification	Uses fuzzy logic & confidence ratings to handle response deviations.
Httprecon	Server behavior analysis	Sends multiple HTTP requests & matches responses with Key Analysis Indexes (KAI).
Wappalyzer	Web technology detection	Open-source; identifies CMS, frameworks, & libraries via HTML analysis.
Aquatone	Reconnaissance & fingerprinting	Uses Wappalyzer's engine for subdomain discovery & webpage similarity detection.
WhatWeb	Web technology identification	Uses regex, fuzzy logic, & webpage metadata (e.g., emails, modules).
FavFreak	Web server fingerprinting	Hashes favicon files for server identification.
Nikto	Web server vulnerability detection	Scans files, directories & analyzes favicon files for security flaws.

Table VII. Various fingerprint tools

HTTPFuzz Framework

Traditional methods of web server fingerprinting frequently have difficulty accurately determining server technologies, particularly when small modifications or custom settings are applied [12]. To address these challenges, HTTPFuzz offers an automated, multi-step system aimed at improving server identification through the use of fuzzing techniques.

Core Components of HTTPFuzz:

Component	**Function & Purpose**
Request Fuzzing Engine	Generates a wide range of varied and modified HTTP request patterns automatically to elicit unique responses from the server
Intelligent Analysis Engine	Utilizes machine learning techniques to examaine server replies and improve the selection process for effective test scenarios
Fingerprinting Engine	Sends well-optimized requests to accurately identify particular web server technologies and setups.

Table VIII. HTTPFuzz framework for fuzzing techniques

XIII. Open for Hire – Trends in Attacks and Misconfiguration Challenges in IoT Devices

This research explores the security vulnerabilities associated with improperly configured Internet of Things (IoT) devices, which attackers frequently target due to easily guessable passwords, factory settings, and insecure communication methods [7] Key Findings: Approximately 1.8 million vulnerable IoT devices were identified through extensive internet scanning. Over the course of a month, six sophisticated honeypots documented 200,209 attempts at attacks, which included denial-of-service and multi-stage strategies. An examination of 81 billion network requests indicated the improper use of IoT protocols. 11,118 misconfigured devices were actively participating in attacks against other systems.

Detection Methods:

Broad Internet Scanning: Focus on identifying vulnerable protocols such as MQTT, CoAP, AMQP, XMPP, UPnP, and Telnet. Tools utilized include: ZMap, ZGrab, and bespoke scripts. Utilization of Public Datasets: Validated findings through cross-referencing data from sources like Shodan and Project Sonar.

Identification Techniques:

- **Banner Evaluation:** Gathered metadata from device banners.
- **Response Evaluation:** Examined device reactions for absent security features such as authentication and encryption.

The study underscores the extensive vulnerability of IoT devices caused by inadequate configuration methods. It emphasizes the critical necessity for enhanced IoT security protocols, such as improved authentication, secure communication, and the removal of default settings to avert potential exploitation.

Analysis of Honeypot and Network Telescope Data

Category	Findings
Malicious IP Addresses	11,182 misconfigured IoT device IPs flagged as malicious by VirusTotal.
Identified IoT Devices	1,671 devices found via Censys, primarily cameras, routers, and IP phones.
Non-IoT Attack Sources	797 registered domains, 427 with active websites.
Suspicious IP Subnets	Unused IPs in /30 and /29 subnets, potential sources of Telnet malware injections.
Malicious URLs Identified	346 URLs linked to WordPress sites, Ubuntu Apache test pages, static ad pages, and fake online stores.

Table XI. Study of different honeypot

XIV. IoT –Honeypot Fingerprinting

The research explores how to identify honeypots in IoT settings to avoid distorted security assessments arising from configuration errors. In April 2021, researchers set up a laboratory to deploy different open-source IoT honeypots, [14] simulating authentic attack scenarios. By analyzing actual scanning data, the investigation examined more than 1.8 million IoT devices exposed online. Utilizing banner-based fingerprinting methods and traffic evaluation from a /8 network telescope (which processed 81 billion requests), the team could differentiate between genuine devices and honeypots. Moreover, they identified 11,118 misconfigured IoT devices that were involved in attacks, alongside 200,209 recorded attacks on honeypots within a single month. These results highlight prevalent protocol misuse (MQTT, CoAP, UPnP) and vulnerabilities in IoT systems [13]. The study emphasizes the need for enhanced fingerprinting precision, broader protocol coverage, and more thorough traffic analysis to strengthen IoT security in practical scenarios.

CONCLUSION

This study offers an in-depth analysis of cybersecurity issues within various digital sectors, including authentication methods, fingerprinting strategies, vulnerabilities of IoT devices, and misconfigurations of cloud services. By utilizing a diverse approach, the research emphasizes the essential role of cryptographic best practices—such as salting, hashing, and iterative encryption—in strengthening outdated authentication mechanisms. It also introduces sophisticated fingerprinting tools and methodologies, like HTTPFuzz and Dossier, to enhance server identification and forensic examination. Additionally, the results uncover the pervasive security weaknesses resulting from improperly configured IoT devices and cloud platforms, highlighting the urgent need for proactive risk-reduction tactics. In summary, this research stresses the intricacy and interconnected nature of contemporary cyber threats, promoting the need for scalable, cooperative, and adaptive defense strategies to bolster global cybersecurity resilience.

REFERENCES

[1] S. Karod, N. Sharma, & A. Sharma. (2015, December). An improved hashing based password security scheme using salting and differential masking. In *2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO)* (pp. xx–xx). IEEE.

[2] Krebs, B. (2016, December). My Yahoo account was hacked! Now what? *Krebs on Security*. https://krebsonsecurity.com/2016/12/my-yahoo-accountwas-hacked-now-what/

[3] Graupner, H., Jaeger, D., Cheng, F., & Meinel, C. (2016, May). Automated parsing and interpretation of identity leaks. In *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security*. Hasso Plattner Institute, University of Potsdam.

[4] Spring, T. (2017, January). Hello Kitty database of 3.3 million breached credentials surfaces. *Threat post*. https://threatpost.com/hello-kitty-database-of3-3-million-breached-credentials-surfaces/122932/

[5] Strahs, B., Yue, C., & Wang, H. (2009). Secure passwords through enhanced hashing. *The College of William and Mary*.

[6] Kristol, D., Gabber, E., Gibbons, P., Matias, Y., & Mayer, A. (1998, June). Design and implementation of the Lucent Personalized Web Assistant (LPWA). *Bell Laboratories, Lucent Technologies*.

[7] Ross, B., Jackson, C., Maiyake, N., Boneh, D., & Mitchell, J. (2005). Stronger password authentication using browser extensions. *Department of Computer Science, Stanford University*.

[8] Halderman, J., Waters, B., & Felten, E. (2005). A convenient method for securely managing passwords. *Princeton & Stanford Universities*.

[9] Bates, A., Tian, D., Butler, K. R. B., & Moyer, T. (2015). Trust worthy whole-system provenance for the Linux kernel. In *USENIX Security Symposium*.

[10] Lee, K. H., Zhang, X., & Xu, D. (2013). High accuracy attack provenance via binary-based execution partition. In *Network and Distributed System Security Symposium (NDSS)*.

[11] Ma, S., Zhang, X., & Xu, D. (2016). ProTracer: Towards practical provenance tracing by alternating between logging and tainting. In *NDSS*.

[12] Sultana, S., Bertino, E., & Shehab, M. (2011). A provenance-based mechanism to identify malicious packet dropping adversaries in sensor networks. In *31st International Conference on Distributed Computing Systems* (pp. 332–338).

[13] Jabiyev, B., Sprecher, S., Onarlioglu, K., & Kirda, E. (2021). T-Reqs: HTTP request smuggling with differential fuzzing. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security* (pp. 1805–1820).

[14] Babun, L., Denney, K., Celik, Z. B., McDaniel, P., & Uluagac, A. S. (2021). A survey on IoT platforms: Communication, security, and privacy perspectives. *Computer Networks*, *192*, 108040. https://doi.org/10.1016/j.comnet.2021.108040

[15] Vignau, B., Khoury, R., Hallé, S., & Lhadj, A. H. (2021). The evolution of IoT malwares, from 2008 to 2019: Survey, taxonomy, process simulator and perspectives. *Journal of Systems Architecture, 116*, 102143. https://doi.org/10.1016/j.sysarc.2021.102143

JETIR