VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama ", Belgaum -590 018, Karnataka State, India

A PROJECT REPORT

ON

"RANSOMWARE READINESS ASSESSMENT TOOL"

Submitted on partial fulfillment of academic requirement for the academic year

2023-24

BACHELOR OF ENGINEERING

IN

INFORMATION SCIENCE AND ENGINEERING

Submitted by

G SHIVARAME GOWDA	**1SJ20IS029**

Under the guidance of

Dr. Nandini S
Associate Professor
Department of ISE, SJCIT

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

SRI JAGADAGURU CHANDRASHEKARANATHA SWAMIJI

INSTITUTE OF TECHNOLOGY

CHICKABALLAPUR – 562 101

2023-2024

|| Jai Sri Gurudev ||

Sri Adichunchanagiri Shikshana Trust ®

SRI JAGADAGURU CHANDRASHEKARANATHA SWAMIJI INSTITUTE OF TECHNOLOGY,

CHICKBALLAPUR-562101 (KARNATAKA)

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project work entitled **"RANSOMWARE READINESS ASSESSMENT TOOL"** is a bonafide work carried out by **G SHIVARAME GOWDA (1SJ20IS029)** in partial fulfillment for the award of **Bachelor of Engineering in Information Science and Engineering in Eighth semester of the Visvesvaraya Technological University, Belagavi** during the year **2023-24**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the department library. The project report has been approved as it satisfies the academic requirements prescribed for the Bachelor of Engineering degree.

....

Signature of the Guide
Dr. Nandini S
Associate Professor,
Department of ISE,SJCIT

....

Signature of the HOD
Mr. Satheesh Chandra Reddy
Professor and HOD
Department of ISE,SJCIT

....

Signature of the Principal
Dr. G T Raju
Principal
SJCIT

Name and signature of the Examiners

Name **Signature**

1. **…………………………………………… ………………………**

2. **……………………………………………. ………………………**

DECLARATION

I G SHIVARAME GOWDA (1SJ20IS029) student of 8th semester BE, S.J.C Institute of Technology, hereby declare that the project entitled **"RANSOMWARE READINESS ASSESSMENT TOOL"** has been carried out by me under the supervision of internal guide **Dr. Nandini S** **Department of ISE** submitted in partial fulfillment of the requirement of the award in the degree of **Bachelor of Engineering in Information Science and Engineering by the Visvesvaraya Technological University** during the academic year 2023-2024. This report has not been submitted to any other organization or University for any award of degree or certificate.

Place :

Date:

G SHIVARAMEGOWDA

ABSTRACT

Ransomwares have become a growing threat in recent years, and this situation continues to worsen. It rose awareness on a particular class of malwares which extort a ransom in exchange for a captive asset. Most widespread ransomwares make an intensive use of data encryption. Basically, they encrypt various files on victim's hard drives, removable drives and mapped network shares before asking for a ransom to get the files decrypted. In this paper, at first, we propose a comprehensive ransomware taxonomy. Then, based on this taxonomy and according to a principal feature which we discovered in high survivable ransomwares (HSR) in the key exchange protocol step, we present a novel approach for detecting high survivable ransomwares and preventing them from encrypting victim's data. Experimental evaluation demonstrates that our framework can detect variants of recent dangerous ransomwares. Cybercriminals and malware writers have diversified their efforts to make money from their victims, using methods that have been well-established on desktops, laptops, tablets and mobile devices, this includes ransomware. We present a novel approach for detecting high survivable ransomwares and preventing them from encrypting victim's data. Our proposed framework is able to detect all current HSRs (High survivable ransomware) before the encryption process starts, thus thwarting and completing the operation completely.

ACKNOWLEDGEMENT

With reverential pranam, I express my sincere gratitude and salutations to the feet of his holiness **Paramapoojya Jagadguru Byravaikya Padmabhushana Sri Sri Sri Dr.Balagangadharanatha MahaSwamij**i, his holiness **Paramapoojya Jagadguru Sri Sri Sri Dr. Nirmalanandanatha Maha Swamiji,** and **Sri Sri MangalnathaSwamiji** of Sri Adichunchanagiri Mutt for their unlimited blessings.

First and foremost, I wish to express our deep sincere feelings of gratitude to our institution, **Sri Jagadguru Chandrashekaranatha Swamiji Institute of Technology,** for providing us an opportunity for completing the Project work successfully.

I extend deep sense of sincere gratitude to **Dr. G T Raju, Principal, S J C Institute of Technology, Chickkaballapur,** for providing an opportunity to complete the Project Work.

I extend special in-depth, heartfelt, and sincere gratitude to **HOD Satheesh Chandra Reddy S, Head of the Department, Information Science and Engineering, S J C Institute of Technology, Chickkaballapur,** for his constant support and valuable guidance of the Project Work.

I convey our sincere thanks to **Project Guide Dr. Nandini S, Associate Professor, Department of Information Science and Engineering, S J C Institute of Technology,** for her constant support, valuable guidance and suggestions of the Project Work.

I also feel immense pleasure to express deep and profound gratitude to **Project Coordinator Prof.Aravinda Thejas Chandra, Associate Professor, Department of Information Science and Engineering, S J C Institute of Technology,** for their guidance and suggestions of the Project Work.

Finally, I would like to thank all faculty members of Department of Information Science and Engineering, S J C Institute of Technology, Chickkaballapur for their support.

I also thank all those who extended their support and co-operation while bringing out this Project Work.

G SHIVARAME GOWDA (1SJ20IS029)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER-1

INTRODUCTION

1.1 OVERVIEW:

Malware attacks continue to remain one of the most popular attack vectors in the world. Compared to other types of malware, ransomware has recently become very popular among malware authors. Ransomware is a kind of scareware that locks a victim's computer until she makes a payment to re-gain access to her data. In fact, this class of malware is not a new concept (i.e., such attacks have been in the wild since the last decade), but the growing number of high-profile ransomware attacks has resulted in increasing concern on how to defend against this class of malware. In 2016, several public and private sectors, including the healthcare industry, were impacted by ransomware [4]. Very recently, WannaCry, one of the successful ransomware attacks, impacted thousands of users around the world by exploiting the EternalBlue vulnerability, encrypting user data, and demanding a bitcoin payment in exchange for unlocking files. While there has been some progress in identifying ransomware attacks, in practice, the primary defence mechanisms to detect, analyse, and defend against ransomware attacks are not very different from the detection techniques that are being used to identify other types of evasive malware attacks. Perhaps the main reason is that this type of malware, similar to other classes of malware, employs common evasion techniques to bypass known detection techniques, reach end-users, and successfully launch attacks. While this is a valid assumption about employing general evasion techniques, the current defence mechanisms cannot achieve the best detection results as evidenced by the increasing number of very successful ransomware attacks in the wild.

1.2 PROBLEM DOMAIN:

The project domain for a Ransomware Readiness Assessment tool encompasses various aspects of cybersecurity, risk management, and compliance, all aimed at helping organizations reduce their vulnerability to ransomware attacks and respond effectively if one occurs. A Ransomware Readiness Assessment tool typically falls within the domain of cybersecurity and risk management. Here are some key aspects of the project domain for a Ransomware Readiness Assessment tool:

- **Cybersecurity**: The primary focus of the tool is to assess an organization's preparedness and resilience against ransomware attacks, which are a subset of cybersecurity threats. This includes evaluating the organization's security measures, policies, and practices.

- **Risk Management**: Ransomware attacks pose a significant risk to organizations, both in terms of data loss and financial impact. The tool is designed to help organizations identify and manage these risks effectively.

- **Information Security**: It covers aspects of information security, such as data protection, access control, encryption, and secure communication, as these are critical components in defending against ransomware attacks.

1.3 PROBLEM STATEMENT:

The increasing sophistication and prevalence of ransomware attacks on Microsoft Windows systems pose a significant threat to data security and system integrity. Traditional malware detection systems, including anti-virus software, often struggle to effectively differentiate between benign software, generic malware, and the distinct characteristics of ransomware in real-time.

Most widespread ransomwares make an intensive use of data encryption. Basically, they encrypt various files on victim’s hard drives, removable drives and mapped network shares before asking for a ransom to get the files decrypted. We present a novel approach for detecting high survivable ransomwares and preventing them from encrypting victim’s data. Our proposed framework is able to detect all current HSRs (High survivable ransomware) before the encryption process starts, thus thwarting the operation completely.

Challenges:

- **Rising Ransomware Threats:** With the surge in ransomware attacks, there is a critical need for robust and accurate detection mechanisms to safeguard Windows systems from potential compromise.

- **Detection Ambiguity:** Existing malware detection tools may face challenges in distinguishing ransomware from other forms of malware due to overlapping characteristics, leading to false positives or undetected ransomware instances.

- **Dynamic Ransomware Variants:** The evolving nature of ransomware, with attackers frequently modifying their tactics and techniques, demands adaptive detection methods capable of identifying new and unknown variants.

1.4 OBJECTIVES:

- Development of DGA-detector, and a novel monitoring framework called CM&CB to detect and prevent damage by the most dangerous ransomwares.
- Presenting a novel approach framework for detecting high survivable ransomwares and preventing them from encrypting victim’s data.
- Design the tool to conduct a comprehensive assessment of an organization's current state of readiness to combat ransomware attacks. This should encompass various aspects of cybersecurity, including policies, procedures, technical safeguards, and employee training
- Ensure that the tool itself is secure and respects user privacy. Data collected during assessments should be handled with care and stored securely.

CHAPTER-2

LITERATURE SURVEY

Table 2.1: Literature Survey

Sl.No	Literature Paper	Methodology Used	Advantages	Disadvantages
[1]	Comparative analysis of various ransomware virii	Comparative analysis of various ransomware virii using reverse engineering approach.	Proposed approach works well on symmetric crypto systems	This method fails to work on hybrid cryptosystems.
[2]	Detecting Algorithmically Generated Domain-FluxAttacks With DNS Traffic Analysis	Proposed methodology to detect such “domain fluxes” in DNS traffic by looking for patterns inherent to domain names that are generated algorithmically	Crawling of all ipv4 domains	Domains which are using IPV6 address cannot exploited in this approach
[3]	Real Time Android Ransomware Detectionby Analyzed Android Applications	By analyzing the Json files of the Apks , patterns of the ransomware behaviors will be identified	Works well in all android applications	Static analysis approach is used, for new ransomwares identification this approach fails
[4]	A Multi-Classifier Network-Based CryptoRansomware DetectionSystem	This paper demonstrates a comprehensive behavioral analysis of crypto ransomware network activities	Early detection Reduced false positives Improved coverage Scalability	Complexity Computational overhead Vulnerability to adversarial attacks

Table 2.1: Literature Survey

Sl. No	Literature Paper	Methodology Used	Advantages	Disadvantages
[5]	Ransomware detection by mining API call usage	Application Programming Interface (API) calls are extracted from the executables and the most discriminating API calls are used to train a classifier to detect unknown ransomware	Authors have tested this method on various classifiers like Decision trees, KNN, Random forest	Solely API mining is not enough stable to identify diverse ransomwares
[6]	A Content-Based Ransomware Detection and Backup Solid-State Drive for Ransomware Defense	This approach uses hardware accelerator to run content-based detection algorithms for ransomware detection at high speed	Device-level backup solution that does not require additional storage for backup	backup solutions can be under the control of ransomware and backup copies can be destroyed by ransomware
[7]	API Call Based Ransomware Dynamic Detection Approach Using TextCNN	This paper proposes a Dynamic Ransomware Detector based on the improved TextCNN(DRDT)	DRDT is trained with ransomware and benign software's API call sequences	Though the approach is effective still it is vulnerable since it allows unknown programms to communicate with DRDT.
[8]	An empirical study of ransomware attacks on organizations: an assessment of severity and salient factors affecting vulnerability	Data was analysed using statistical tests, such as Fisher's Exact tests, to assess the severity of the attacks and examine the influence of various factors	Provides real-world data and insights Identifies factors affecting vulnerability	Limited scope Data collection challenges

CHAPTER-3

SYSTEM ANALYSIS

3.1 PROBLEM IDENTIFICATION:

Ransomware attacks have become a pervasive threat in today's digital landscape. These malicious attacks encrypt files or lock users out of their systems until a ransom is paid, often in cryptocurrency. The consequences can be catastrophic, leading to data loss, financial damage, and reputational harm for businesses and organizations. To mitigate this threat, many entities employ ransomware readiness assessment tools. However, the effectiveness of these tools can vary, and understanding the challenges in identifying and addressing ransomware risks is crucial.

3.2 PROBLEM DEFINITION:

The problem definition of a Ransomware Readiness Assessment Tool involves identifying and evaluating an organization's preparedness to defend against, respond to, and recover from ransomware attacks. This tool aims to address the growing threat of ransomware by providing a structured assessment framework. The key challenges include the need to understand the organization's current security posture, identify vulnerabilities, assess incident response capabilities, and establish effective recovery mechanisms. Additionally, the tool should consider factors such as employee awareness, security policies, technology infrastructure, and data backup strategies. Ultimately, the goal is to help organizations proactively mitigate the risks associated with ransomware attacks and minimize the potential impact on operations and data integrity.

3.3 EXISTING SYSTEM:

Ransomware readiness assessment tool is designed to evaluate an organization's preparedness to defend against and respond to ransomware attacks. The problem at hand is that many organizations lack a comprehensive and systematic approach to assess their readiness and resilience against ransomware threats.

3.3.1 DISADVANTAGES OF EXISTING SYSTEM:

- **Lack of Awareness**: Many organizations underestimate the severity and frequency of ransomware attacks, leading to a lack of awareness regarding the need for robust ransomware readiness measures.
- **Inadequate Preparedness**: Organizations often do not have well-defined strategies, policies, and practices in place to prevent ransomware attacks, detect them promptly, and respond effectively.
- **Insufficient Training**: Employees may not be adequately trained and educated on cybersecurity best practices, making them susceptible to inadvertently facilitating ransomware attacks.
- **Outdated Technology**: Legacy systems and outdated security measures may leave organizations vulnerable to ransomware attacks, as attackers continually evolve their techniques.

3.4 PROPOSED SYSTEM:

To address these challenges and mitigate the risks associated with ransomware attacks, there is a pressing need for a Ransomware Readiness Assessment tool. This tool will provide organizations with a structured and holistic approach to evaluate their current cybersecurity posture, identify vulnerabilities, and develop strategies to enhance ransomware resilience. By conducting regular assessments and implementing recommended improvements, organizations can reduce the likelihood of falling victim to ransomware attacks and minimize the impact if an attack does occur. Ultimately, this tool seeks to empower organizations to proactively protect their critical data, operations, and reputation against the growing menace of ransomware attacks.

3.4.1 ADVANTAGES OF PROPOSED SYTEM:

- Identifying and prioritizing vulnerabilities in their cybersecurity defences.
- Evaluating the effectiveness of employee training and awareness programs.
- Ensuring compliance with relevant cybersecurity regulations and standards.
- Assessing the robustness of data backup and recovery processes.
- Enhancing incident response capabilities to minimize downtime and data loss.
- Monitoring and adapting to evolving ransomware threats through threat intelligence.

CHAPTER-4

SYSTEM REQUIREMENT SPECIFICATION

4.1 FUNCTIONAL REQUIREMENTS:

- **File Input Handling:** The system must accept Windows executable files in the PE file format, specifically those with '.exe' extensions, as input for analysis.
- **Ensemble Model:** Implement two deep learning neural network classifiers and combine them into an ensemble for ransomware detection.
- **Classification Output:** The system should provide a clear classification output for each input file, categorizing them as benign, generic malware, or ransomware.
- **Integration with Anti-Virus Software:** Enable seamless integration with existing anti-virus software or malware detection systems to extend and enhance their capabilities.
- **Automation of Static Analysis:** Automate the static analysis of Windows executable files, extracting relevant features and abstracting patterns within these features to facilitate the classification process.

4.2 NON-FUNCTIONAL REQUIREMENTS:

- **Accuracy:** The system must achieve a high level of accuracy in classifying files, with a focus on minimizing false positives and false negatives.
- **Performance:** Ensure that the system's performance meets acceptable response times, even with a large number of concurrent requests for analysis.
- **Security:** Implement robust security measures to safeguard the system from potential attacks, ensuring the confidentiality and integrity of classified data.
- **Reliability:** The system should be reliable, minimizing downtime and ensuring consistent availability for users.
- **Scalability:** Design the system architecture to scale efficiently with increasing demands and data volumes without compromising performance.

4.3 HARDWARE REQUIREMENTS:

- Processor: Intel CoreTM – i5
- Speed: 2.4 GHZ
- RAM: 8 GB RAM
- Hard disk: 80 GB HDD

4.4 SOFTWARE REQUIREMENTS:

- Operating System: Windows 64-bit
- Technology: Python
- IDE: Pycharm
- Python Version: Python 3.6 onwards

4.5 PACKAGES REQUIREMENTS:

- **NumPy:** For numerical operations and efficient handling of arrays and matrices.
- **Pandas:** Useful for data manipulation and analysis, especially for handling datasets.
- **Matplotlib and Seaborn:** For data visualization and creating plots to analyze the performance of your models.
- **Scikit-learn:** Provides simple and efficient tools for data mining and data analysis. It includes various machine learning algorithms and tools for model evaluation.
- **TensorFlow or PyTorch:** Deep learning frameworks for building and training neural networks. Choose one based on your preference and requirements.
- **Keras:** If you choose TensorFlow as your deep learning framework, Keras is a high-level neural networks API that works seamlessly with TensorFlow.
- **pefile:** A Python module for working with Portable Executable (PE) files, the format used by Windows for executable files.
- **bytecode:** For extracting features from binary files by analyzing their bytecode.
- **Scikit-learn's Preprocessing module:** For scaling, normalizing, and encoding features.
- **Scikit-learn's Metrics module:** For evaluating the performance of your machine learning models using metrics like accuracy, precision, recall, and F1 score.

CHAPTER-5

SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE:

Our proposed framework is able to detect all current HSRs before the encryption process starts, thus thwarting the operation completely. During our analysis, we have dissected the HCR attack protocol which lead us to find an effective feature in HCRs, based on this feature we have designed and implemented our framework.

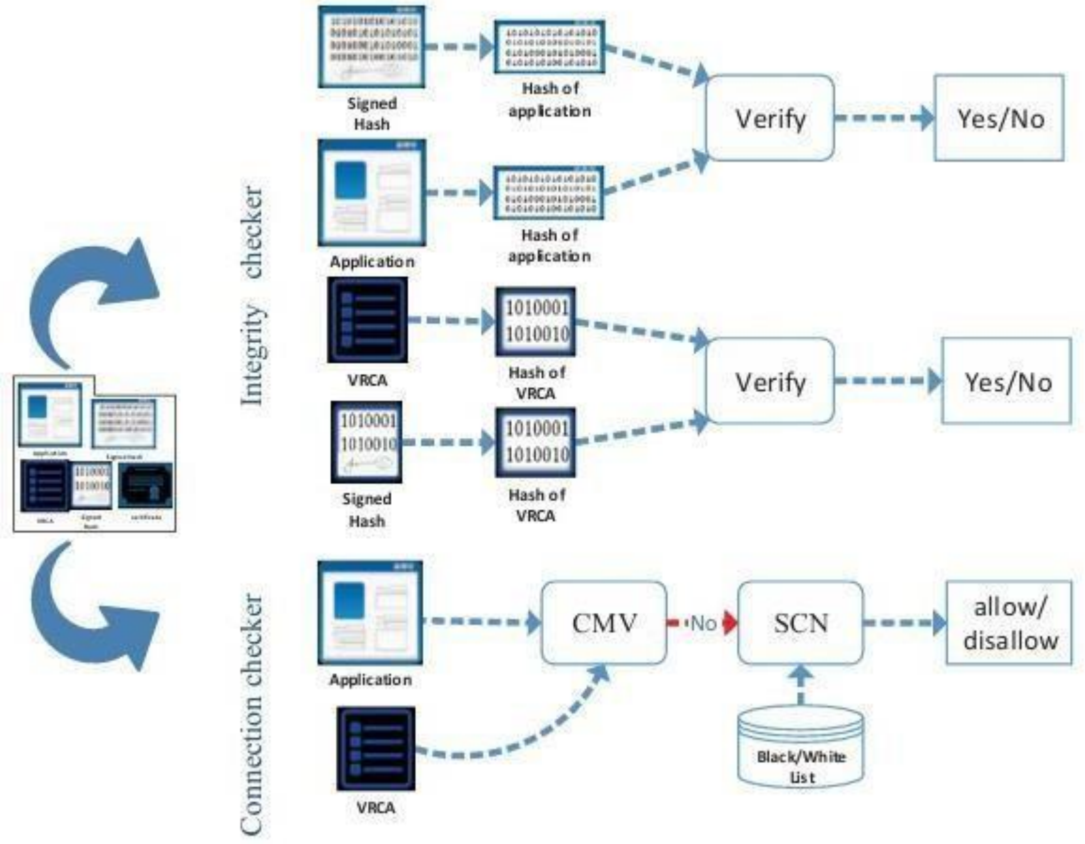


Fig 5.1.1: Architecture of proposed framework

When a ransomware wants to connect to his C&C with DGA the connection monitor verifier (CMV)with the help of DGA detector detects suspicious connection. One another feature which is important in DGA malicious requests is these algorithms generate and request many domains connection in a short time. Then the suspicious connection notifier (SCN) will show the user a suspicious connection and all the user can break the connection and report this suspicious connection address to experts. DGA detector framework beside all the benefits needs a high precision for decreasing false positives. In the other hand some domains are not gibberish sometimes they are in a different language then this framework needs more effort to be a real acceptable popular framework.

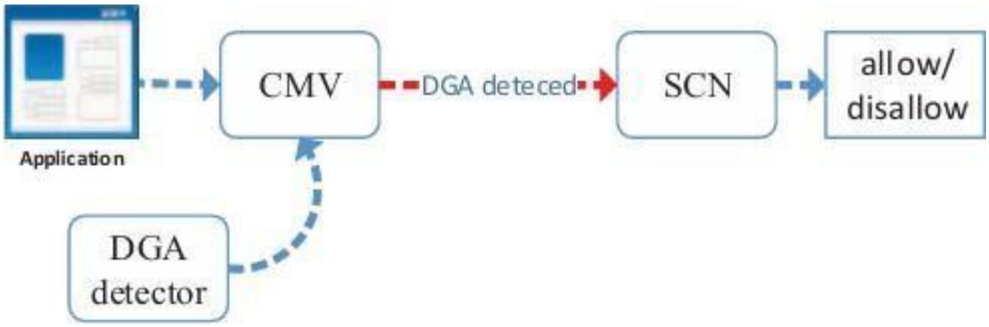


Fig 5.1.2: Architecture of DGA Framework

In the next step we suggest an extended framework based on a same idea which is related to the public key exchange stage in HCR protocol. With this simple but novel idea which is never be proposed for ransomware detection and also malware detection we could successfully detect the current dangerous HCRs such as Cryptolocker, Cryptolocker2, Cryptowall, Cryptowall2 and thwart their encryption process before it startedIn this framework we have implemented a connection-monitor which check all the applications especially the new or untrusted executable files in Windows (as with the majority of ransomware is targeted at Microsoft Windows operating system)

5.2 SHA-256 WORKFLOW:

A secure hashing algorithm or commonly referred to as SHA-256, is unkeyed cryptographic hashing function that takes an input of variable length and produces a 256-bit long hash output. SHA- 256 is one of the first and most prominently used hashing algorithms in blockchains like Bitcoin, Bitcoin Cash, and Bitcoin SV. SHA-256 is used in various stages in a blockchain, most prominently:

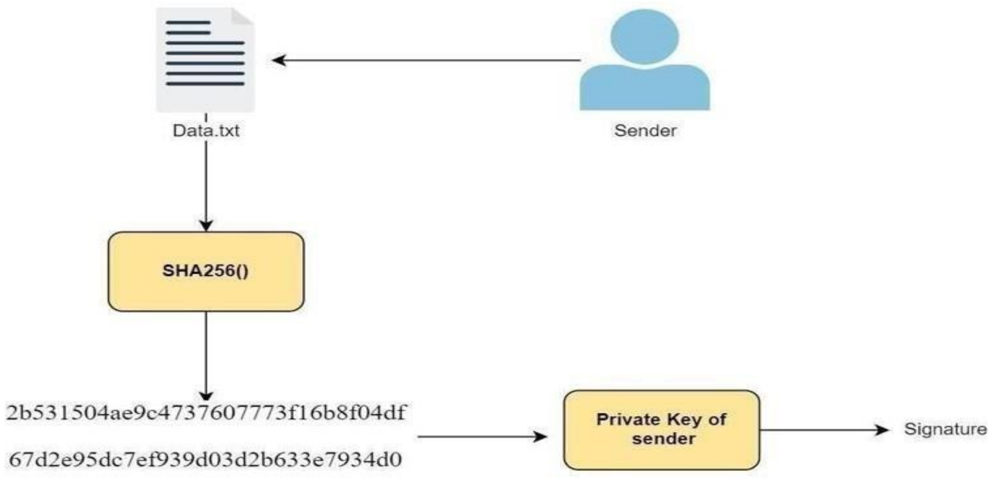

Data.txt
Sender
SHA256()
2b531504ae9c4737607773f16b8f04df
67d2e95dc7ef939d03d2b633e7934d0
Private Key of sender
Signature

Fig 5.2.1: SHA-256 Workflow

Consensus mechanism: Miners calculate the hash of new blocks to be created using SHA-256 by varying the value of nonce in a bitcoin block until they reach the hash below the threshold. Then that block can be accepted into the ledger.

Chains of blocks: Each block in the ledger contains a hash generated by SHA-256 referring to the preceding blocking the chain.

Digital signatures: Transactions use digital signatures to maintain integrity, the information used in the transaction is hashed using SHA-256, and then it is encrypted with the sender's private key to generate a signature. The miner then verifies this signature to validate the transaction. SHA-256 offers security and reliability. Here are some of the main features of SHA-256, which make it perfect to be used as the main hashing function in a blockchain.

Collision resistant: No two input values can produce the same hash output. This ensures that every block in the blockchain ledger is assigned a unique hash value.

Preimage resistance: The input cannot be recreated given a hash value. This ensures that during the proof of work in bitcoin, the miners cannot guess the value of nonce by converting the acceptable hash back into the input; instead, they have to use the brute force method, which ensures that the work is done.

Avalanche effect: If there is a small change in the input, the output changes dramatically. This makes sure that the hash value cannot be guessed based on the input values. This makes the hash more secure.

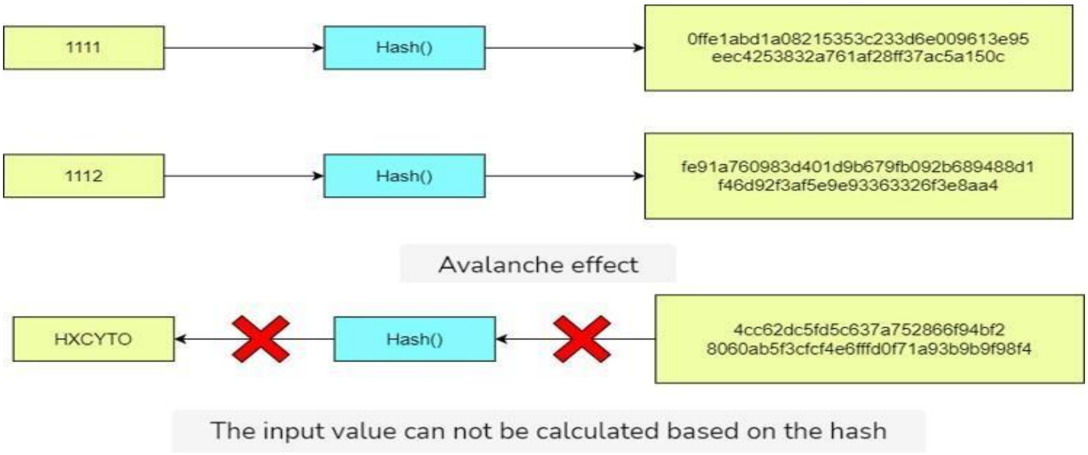

1111
Hash()
0ffe1abd1a08215353c233d6e009613e95
eec4253832a761af28ff37ac5a150c
1112
Hash()
fe91a760983d401d9b679fb092b689488d1
f46d92f3af5e9e93363326f3e8aa4
Avalanche effect
HXCYTO
Hash()
4cc62dc5fd5c637a752866f94bf2
8060ab5f3cfcf4e6fffd0f71a93b9b9f98f4
The input value can not be calculated based on the hash

Fig 5.2.2: Avalanche Effect

5.3 DATAFLOW DIAGRAM:

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFDs can also be used for the visualization of data processing.

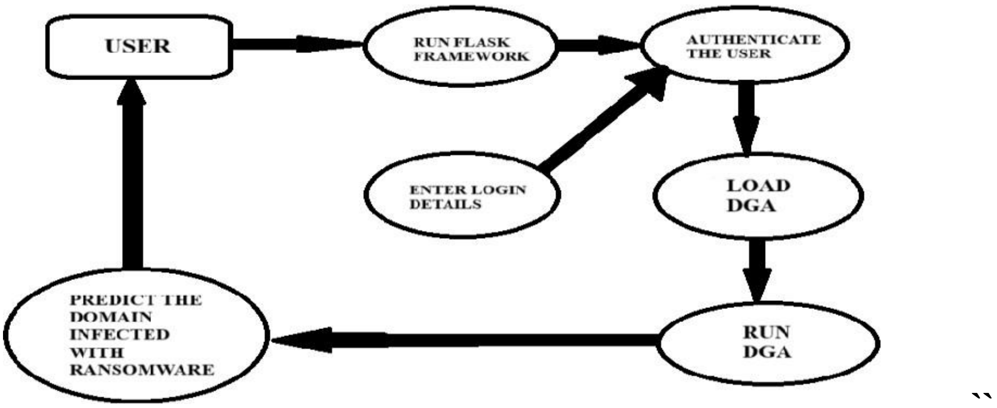


Fig 5.3: Dataflow Diagram

5.4 USE CASE DIAGRAM:

A use case diagram in the Unified Modelling Language (UML) is a type of behavioural diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

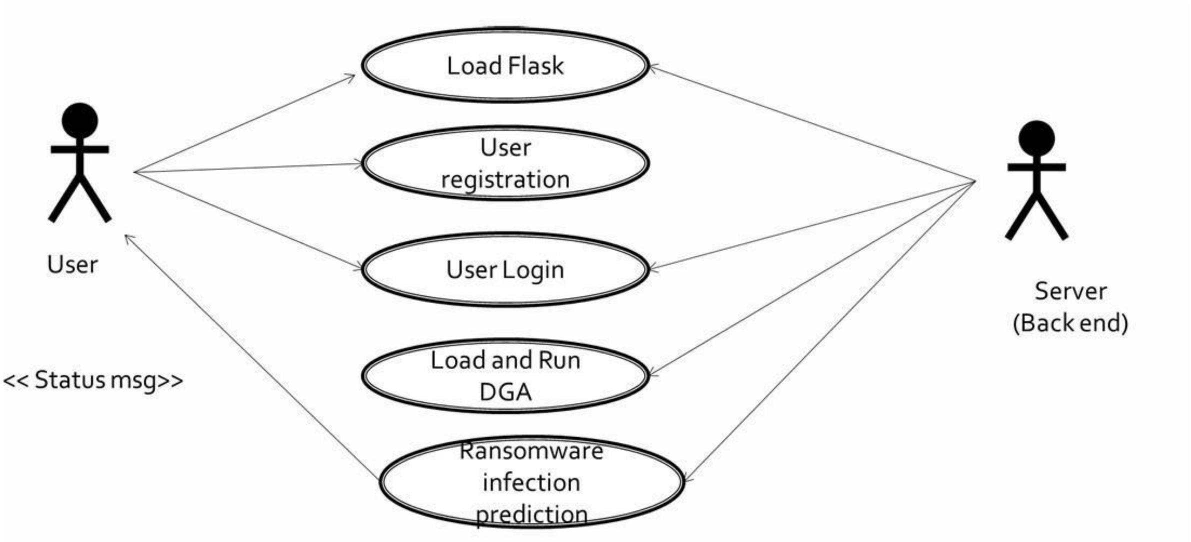


Fig 5.4: Use case Diagram

CHAPTER-6

IMPLEMENTATION

The phase is initiated after the system has been nested and accepted by the user. In this phase, the system is installed to support the intended business functions. System performance is compared to performance objectives established during planning phase. Implementation includes user notification, user training, installation of hardware, installation of software onto production computers, and integration of the system into daily work processes. This phase continues until the system is operating in production in accordance with the defined user requirement Implementation in a project refers to the process of putting plans, designs, or strategies into action to achieve project goals. It involves executing tasks, utilizing resources, and managing timelines to complete project objectives. Implementation is a crucial stage in the project life cycle as it determines project success.

6.1 OVERVIEW OF PYTHON:

Python is a widely used general-purpose, high level programming language. It was created by Guido van Rossum in 1991 and further developed by the Python Software Foundation. It was designed with an emphasis on code readability, and its syntax allows programmers to express their concepts in fewer lines of code. Python is a programming language that lets you work quickly and integrate systems more efficiently.

6.2 LIST OF MODULES:

- User Signup: New user can sign up with the application.
- User Login: After sign up user can login to system.
- Generate & Load DGA Detector: Using this module user can generate and load DGA model.
- Run DGA Detector: Using this module user can enter any domain name and then DGA detector will predict whether domain is Legitimate or Ransomware

6.3 MODULAR DESCRIPTION:

6.3.1 User Signup:

The user signup process is a critical component of any web application, as it's the gateway for users to create accounts and access the platform's features. A well-designed signup process not only ensures security but also provides a smooth and intuitive experience for users.The process typically begins with a registration form, where users input their information such as username, email address, and password. Once the user submits the form, the backend processes the data and stores it securely in the database. This often involves hashing passwords using strong encryption algorithms like bcrypt to protect sensitive user data from unauthorized access.

Pseudocode:

```
Procedure UserSignup(username, password):
Input: username, password
Output: success or failure message
 if username is empty or password is empty:
 return "Username and password are required."
 if username already exists in the database:
return "Username is already taken."
store username and hashed password securely in the database
return "User registration successful."
```

Code:

```
def SignupAction(request):
    if request.method == 'POST':
        username = request.POST.get('t1', False)
        password = request.POST.get('t2', False)
        contact = request.POST.get('t3', False)
        email = request.POST.get('t4', False)
        address = request.POST.get('t5', False)
```

```
status = 'none'
con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'RansomApp',charset='utf8')
with con:
    cur = con.cursor()
    cur.execute("select username from signup where username = '"+username+"'")
    rows = cur.fetchall()
    for row in rows:
        if row[0] == email:
            status = 'Given Username already exists'
            break
if status == 'none':
    db_connection = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'RansomApp',charset='utf8')
    db_cursor = db_connection.cursor()
    student_sql_query = "INSERT INTO signup(username,password,contact_no,email_id,address) VALUES('"+username+"','"+password+"','"+contact+"','"+email+"','"+address+"')"
    db_cursor.execute(student_sql_query)
    db_connection.commit()
    print(db_cursor.rowcount, "Record Inserted")
    if db_cursor.rowcount == 1:
        status = 'Signup Process Completed'
context= {'data':status}
return render(request, 'Signup.html', context)
```

6.3.2 User Login:

The user login process is a critical aspect of web applications, ensuring secure access to user-specific content and functionalities. If the username or email exists and the password matches the stored password hash, the user is considered authenticated. However, storing passwords in plaintext is a security risk, so instead, web applications store hashed passwords. During the login process, the entered password is hashed using the same algorithm and compared with the hashed password stored in the database. If the authentication is successful, the user's identity is established, and the application generates a session for the user. Sessions are a way to persist user data across multiple HTTP requests. A session ID is typically stored in a cookie on the user's browser, allowing the server to recognize and associate subsequent requests with the authenticated user.

Pseudocode:

```
Procedure UserLogin(username, password):
    Input: username, password
    Output: success or failure message
    if username or password is empty:
        return "Please enter username and password."
    retrieve hashed password from the database for the given username
    if username doesn't exist or hashed password doesn't match input password:
        return "Invalid username or password."
    return "Login successful."
```

Code:

```
def UserLoginAction(request):
    if request.method == 'POST':
        global uname
        option = 0
        username = request.POST.get('username', False)
        password = request.POST.get('password', False)
        con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 
'root', database = 'RansomApp',charset='utf8')
        with con:
            cur = con.cursor()
            cur.execute("select * FROM signup")
            rows = cur.fetchall()
            for row in rows:
```

```
            if row[0] == username and row[1] == password:
                uname = username
                option = 1
                break
    if option == 1:
        context= {'data':'welcome '+username}
        return render(request, 'UserScreen.html', context)
    else:
        context= {'data':'Invalid login details'}
        return render(request, 'UserLogin.html', context)
```

6.3.3 Generate and load DGA:

The generation of DGAs requires careful design. Malware authors create algorithms that generate a large number of domain names, often using a combination of random characters, time-based elements, and seed values. These algorithms produce domain names that appear random but are deterministic, meaning the same algorithm and input will always produce the same output. For example, an algorithm might generate domain names based on the current date, a unique identifier, or a combination of both.Loading DGAs into security systems involves integrating them into various tools and platforms used for threat detection and prevention. Security appliances, such as firewalls, intrusion detection systems (IDS), and security information and event management (SIEM) systems, can be configured to recognize and block traffic to and from domains generated by known DGAs.

Pseudocode:

```
Procedure GenerateAndLoadDGADetector():
 Output: success or failure message
 Generate or load the DGA detection model
 if model generation or loading is successful:
 return "DGA detector generated and loaded successfully."
 else:
```

```
 return "Failed to generate or load DGA detector.”
```

Code:

```
def LoadDGA(request):
    if request.method == 'GET':
        global labels, acc
        output = "DGA can detect different Ransomware : "+str(labels)+"<br/>"
        output += "DGA Ransomware Attack Detection Model Loaded<br/>Detection Accuracy % = "+str(acc)
        context= {'data':output}
        return render(request, 'UserScreen.html', context)
def RunDGA(request):
    if request.method == 'GET':
        return render(request, 'RunDGA.html', {})
```

6.3.4 Run DGA Detector:

Running a Domain Generation Algorithm (DGA) is a crucial aspect in cybersecurity, particularly in the realm of threat detection and analysis. DGAs are algorithms used by malware to dynamically generate a large number of domain names that can be used as rendezvous points for command and control (C&C) communication or other malicious activities.Running a Domain Generation Algorithm is a multifaceted process that involves reverse engineering, script development, domain validation, and threat intelligence analysis. It plays a crucial role in cybersecurity operations by enabling the detection, analysis, and mitigation of threats associated with malware employing DGAs.

Pseudocode:

```
Procedure RunDGADetector(domain):
    Input: domain
    Output: prediction result (legitimate or ransomware)
     if domain is empty:
        return "Please enter a domain name."
    pass the domain through the DGA detection model
    if model predicts the domain as legitimate:
        return "Domain is legitimate."
    else:
        return "Domain is associated with ransomware.
```

Preprocess Data**:**

```
dataset=pd.read_csv("Dataset/dga_data.csv",nrows=5000)
labels = np.unique(dataset['subclass'].values.ravel())
dataset = dataset.dropna()
dataset = dataset.values
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2) #split dataset into train and test
X_train, X_test1, y_train, y_test1 = train_test_split(X, Y, test_size=0.1) #split dataset into train and test
```

Train Logistic Regression Model:

```
lr_cls = LogisticRegression(max_iter=300) #create Logistic Regression object
lr_cls.fit(X_train, y_train)
predict = lr_cls.predict(X_test)
acc = accuracy_score(y_test,predict)*100
```

Code:

```
def  RunDGAAction(request):
    if request.method == 'POST':
        global lr_cls, tfidf_vectorizer, sc, labels
        domain = request.POST.get('t1', False)
        vector = tfidf_vectorizer.transform([domain]).toarray()
        vector = sc.transform(vector)
        predict = lr_cls.predict(vector)[0]
        predict = int(predict)
        print(predict)
        predict = labels[predict]
        output = "Given Domain = "+domain+"<br/>"
        output += "DGA Predicted AS ====> "+predict
        context= {'data':output}
        return render(request, 'RunDGA.html', context)
def UserLogin(request):
    if request.method == 'GET':
       return render(request, 'UserLogin.html', {})
def index(request):
    if request.method == 'GET':
       return render(request, 'index.html', {})
def Signup(request):
    if request.method == 'GET':
       return render(request, 'Signup.html', {})
```

CHAPTER-7

TESTING

7.1 SOFTWARE TESTING:

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

7.2 TYPES OF TESTING:

- **Unit testing:**

  Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

- **Integration testing:**

  Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

- **Functional testing:**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centered on the following items:

Valid Input: identified classes of valid input must be accepted.

Invalid Input: identified classes of invalid input must be rejected. Functions : identified functions must be exercised.

Output: identified classes of application outputs must be exercised. Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

- **White Box testing:**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

- **Black Box testing:**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

- **Acceptance testing:**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

7.3 TEST CASES:

Table 7.3: Test Cases

TC No	Test Name	Test Description	Input	Expected Output	Actual Output	Test Result
UTC01	Successful Signup	Test user signup with valid credentials	Username, Password	Success message	Success message	Pass
UTC02	Successful Login	Login with valid credentials	Username, Password	Dashboard access	Dashboard access	Pass
UTC03	Generate Model	Generate DGA detection model successfully	-	Success message	Success message	Pass
UTC04	Load Model	Load existing DGA detection model	Model file path	Success message	Success message	Pass
UTC05	Legitimate Domain	Test a legitimate domain	Legitimate domain name	"Legitimate" prediction	"Legitimate" prediction	Pass
UTC06	Ransomware Domain	Test a known ransomware domain	Ransomware domain name	"Ransomware" prediction	"Ransomware" prediction	Pass

CHAPTER-8

CONCLUSION AND FUTURE ENHANCEMENTS

In this project we presented several techniques to counter the threat caused by dangerous ransomware. The proposed techniques include a DGA-detector, and a novel monitoring framework called CM&CB to detect and prevent damage by the most dangerous ransomware. The key observation of this approach is that the operation of HSRs relies on a key-exchange step. By monitoring and blocking this step, the whole operation of the HSR is thwarted.

The main advantages of the proposed idea can be summarized as following. First, this framework is the first framework which is designed focused on the issue of ransomwares, by monitoring suspicious connections and preventing them from encrypting the victim's data. The experimental evaluations show that the proposed framework can successfully thwart the most dangerous HSRs, which was an open problem in the field of malware mitigation. Currently, further research is taking place on developing and more extensive evaluation. In addition, this framework is also useful in the detection of other types of malicious software, such as Bitcoin- mining malwares, botnets, drive-by download malwares and etc.

Our long term objective is to extend this framework by adding another HSR features to detect new and unknown sophisticated HSRs which they will not be detected with only key-exchange step feature.

BIBLIOGRAPHY

[1] Gazet, Alexandre. "Comparative analysis of various ransomware virii." Journal in computer virology 6.1 (2010): 77-90.

[2] Young, Adam, and Moti Yung. "Cryptovirology: Extortion-based security threats and countermeasures." Security and Privacy, 1996. Proceedings., 1996 IEEE Symposium on. IEEE, 1996.

[3] Shivale, Saurabh Anandrao. "Cryptovirology: Virus Approach." arXiv preprint arXiv:1108.2482 (2011).

[4] "Update: McAfee: Cyber criminals using Android malware and ransomware the most". InfoWorld. Retrieved 16 September 2013.

[5] "Cryptolocker victims to get files back for free". BBC News. 6 August 2014. Retrieved 18 August 2014.

[6] Violet Blue (December 22, 2013). "CryptoLocker's crimewave: A trail of millions in laundered Bitcoin". ZDNet. Retrieved 2013-12-23.

[7] McAfee Threats Report: February 2015, By McAfee Labs,Page 38,2015.

[8] McAfee Threats Report: Third Quarter 2013, By McAfee Labs,Page 19,2013.

[9] McAfee Threats Report: Second Quarter 2014, By McAfee Labs,Page 21 ,2014

[10] Young, Adam, and Moti Yung. Malicious cryptography: Exposing cryptovirology. John Wiley & Sons, 2004.

[11] Adleman, Leonard M. "An abstract theory of computer viruses." Proceedings on Advances in cryptology. Springer-Verlag New York, Inc., 1990.

[12] Abidin, Shafiqul, Rajeev Kumar, and Varun Tiwari. "A Review Report on Cryptovirology and Cryptography." International Journal of Scientific & Engineering Research 3.11 (2012): 1.

[13] Stone-Gross, Brett, et al. "Your botnet is my botnet: analysis of a botnet takeover." Proceedings of the 16th ACM conference on Computer and communications security. ACM, 2009.

[14] Yadav, S., Ashwath K. K. R., and Supranamaya R. . "Detecting algorithmically generated domain-flux attacks with DNS traffic analysis." Networking, IEEE/ACM Transactions on 20.5 (2012): 1663- 1677.

APPENDIX

APPENDIX A: SNAPSHOTS

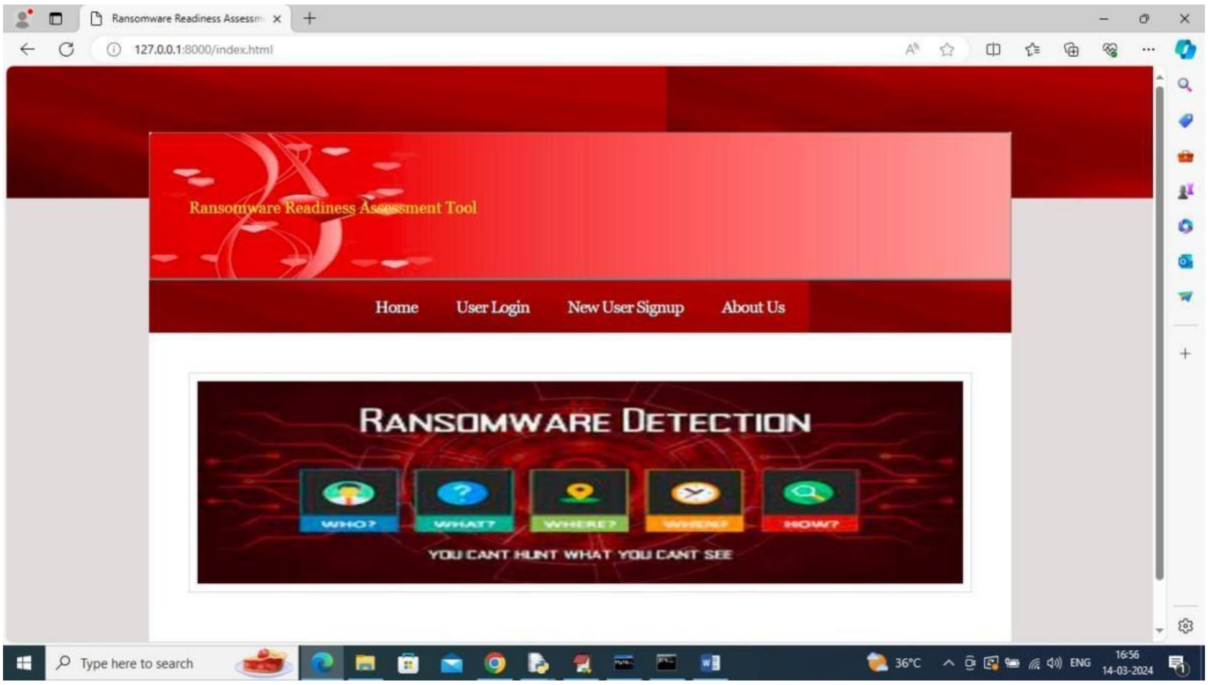
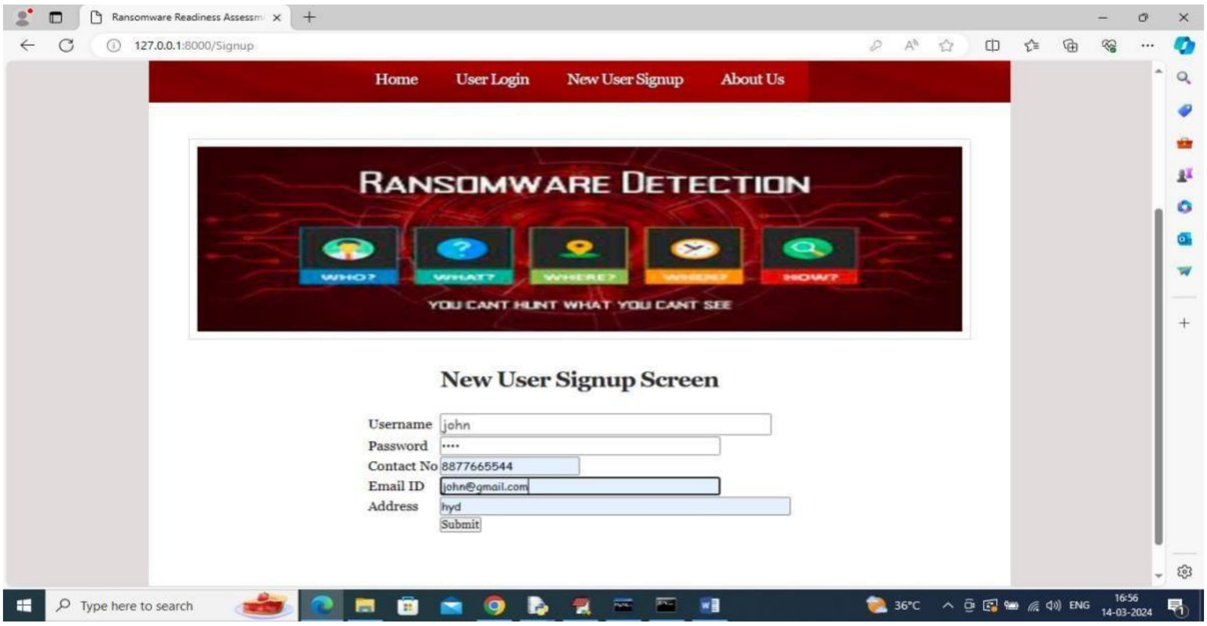


Fig A1: Home Page

Description: In above screen click on "New User Sign up'.

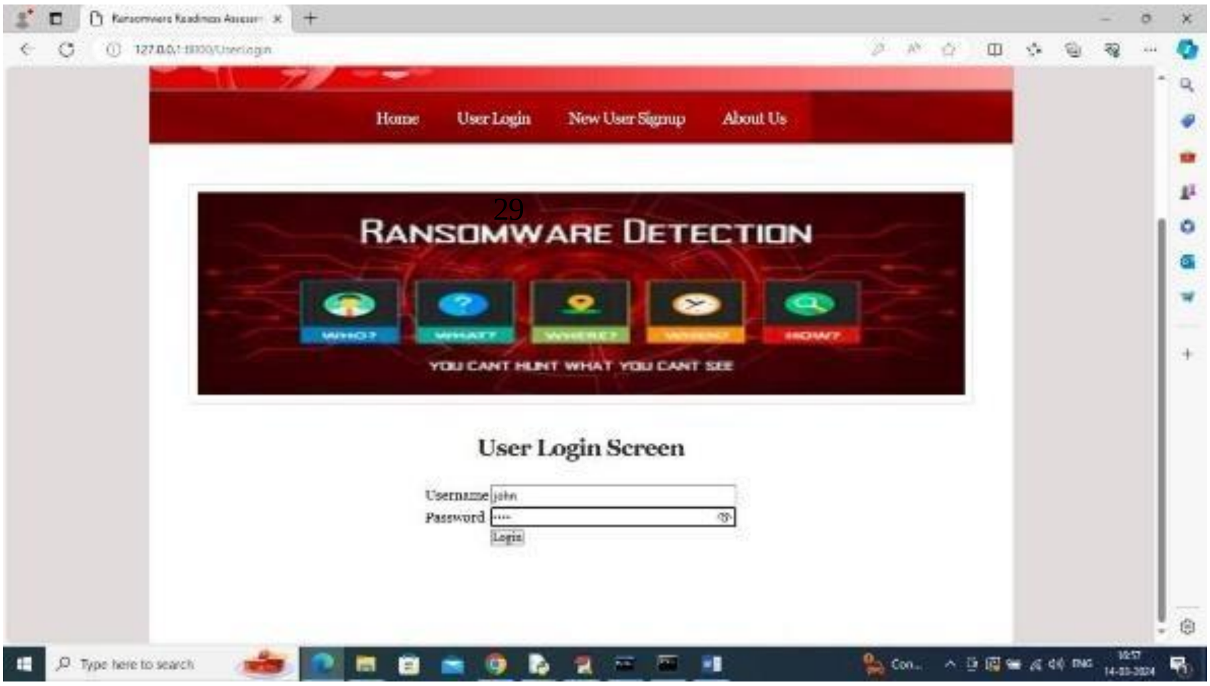


New User Signup Screen

Username john
Password ····
Contact No 8877665544
Email ID john@gmail.com
Address hyd
Submit

Fig A2: User Sign Up Page

Description: In this screen user is entering sign up details.



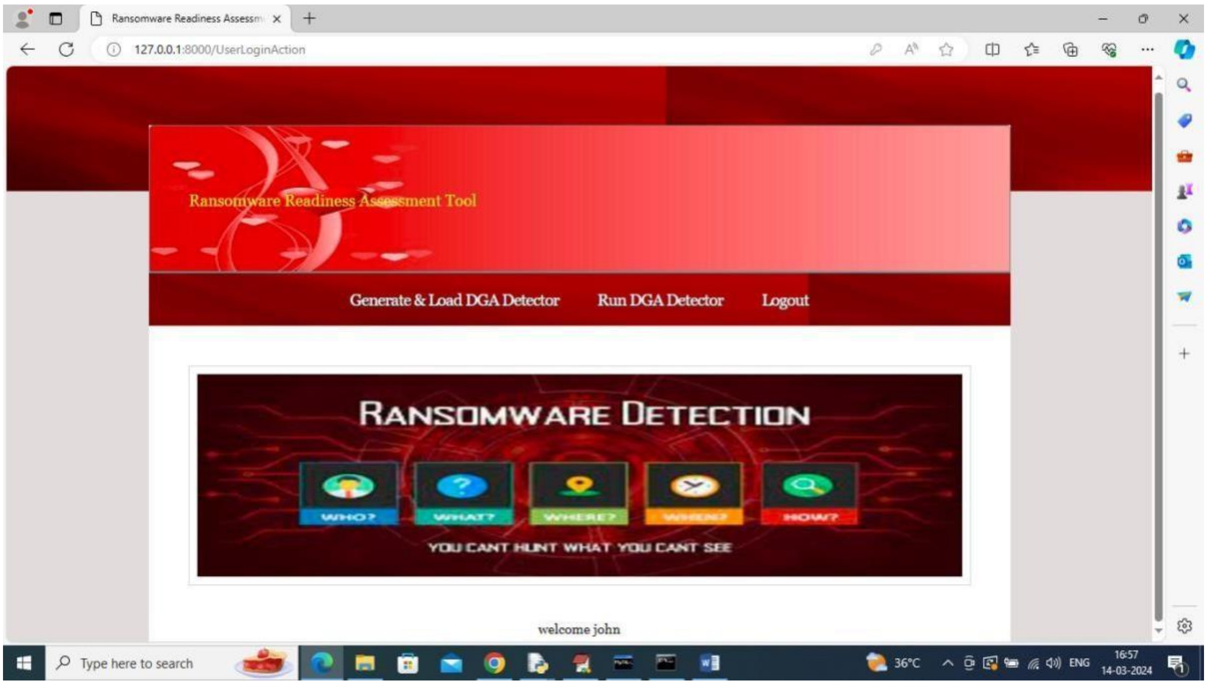
User Login Screen

Username john

Password ····

Fig A3: User Login Page

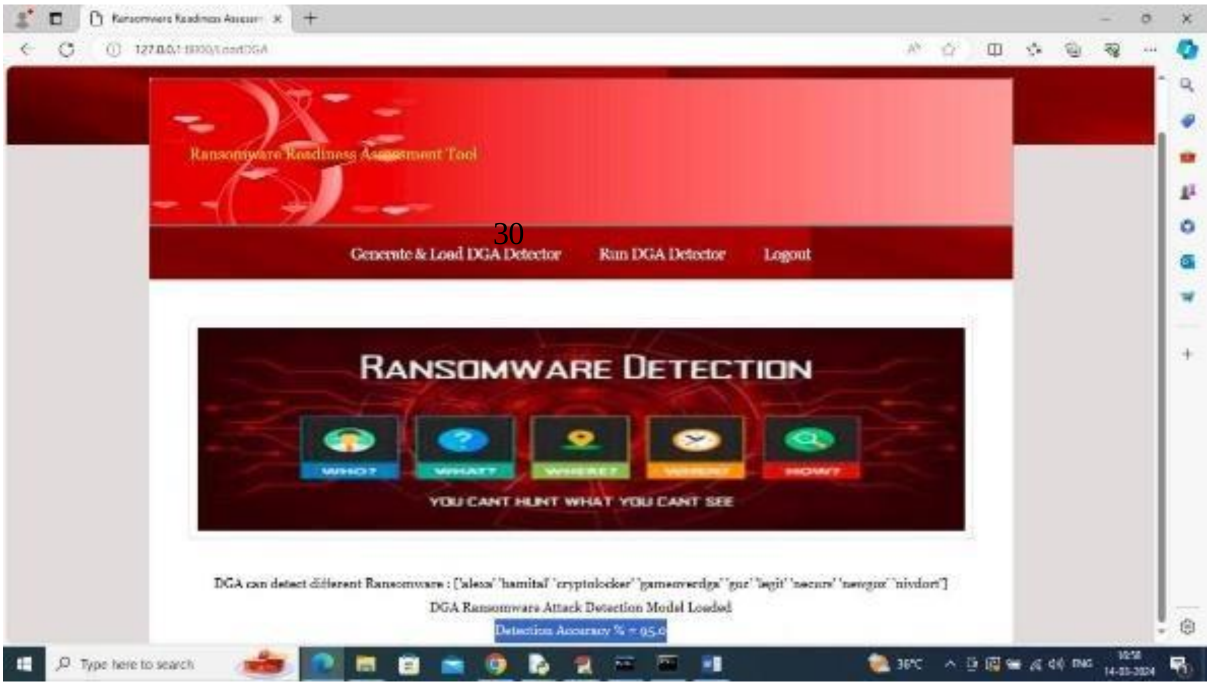
Description: In this screen user is entering their login details.


Ransomware Readiness Assessm
127.0.0.1:8000/UserLoginAction
Ransomware Readiness Assessment Tool
Generate & Load DGA Detector
Run DGA Detector
Logout
RANSOMWARE DETECTION
WHO?
WHAT?
WHERE?
WHEN?
HOW?
YOU CANT HUNT WHAT YOU CANT SEE
Type here to search
36°C
ENG
16:57
14-03-2024

welcome john

Fig A4: Welcome Screen

Description: Welcome screen is displayed after the user login.


Ransomware Readiness Assessment Tool
Generate & Load DGA Detector
Run DGA Detector
Logout
RANSOMWARE DETECTION
WHO?
WHAT?
WHERE?
WHEN?
HOW?
YOU CANT HUNT WHAT YOU CANT SEE

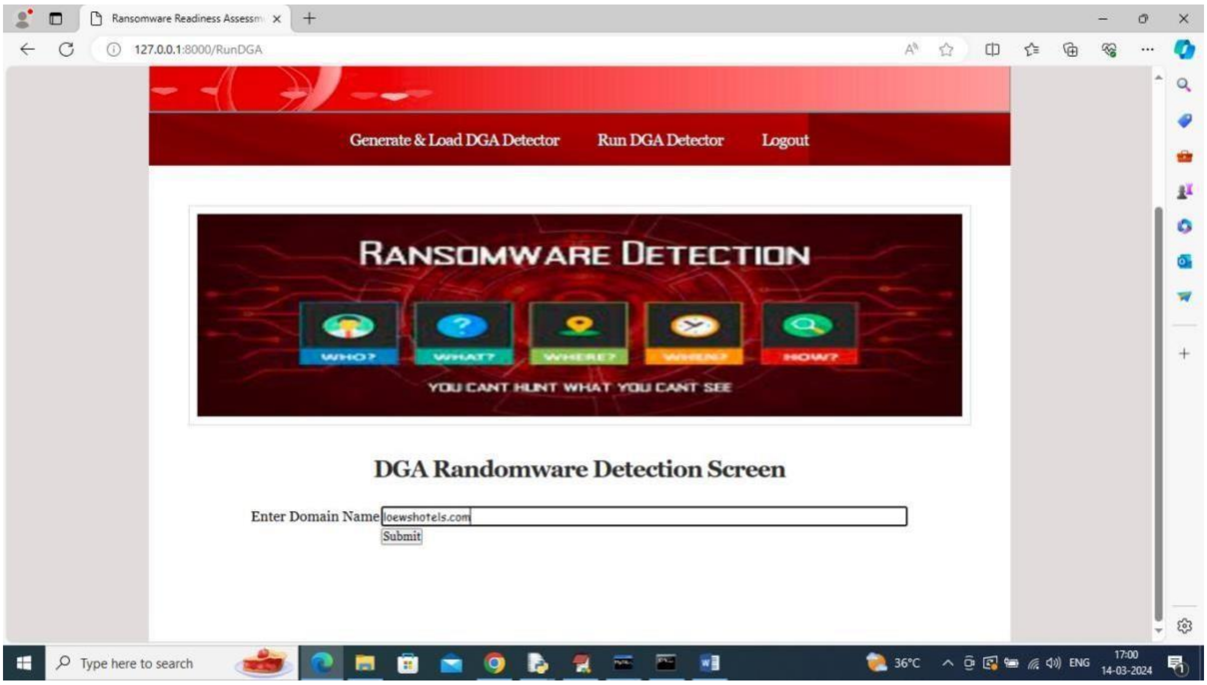
DGA can detect different Ransomware : ['alexa' 'bamital' 'cryptolocker' 'gameoverdga' 'goz' 'legit' 'necurs' 'newgoz' 'nivdort']

DGA Ransomware Attack Detection Model Loaded

Detection Accuracy % = 95.0

Fig A5: Detection Accuracy

Description: In this screen DGA detector detection accuracy is 95% and in first line DGA can detect 9 different Ransomware attacks and those attack names are 'alexa' 'bamital' 'cryptolocker' 'gameoverdga' 'goz' 'legit' 'necurs' 'newgoz' 'nivdort'. In above names ALEXA and LIGIT are normal domains.

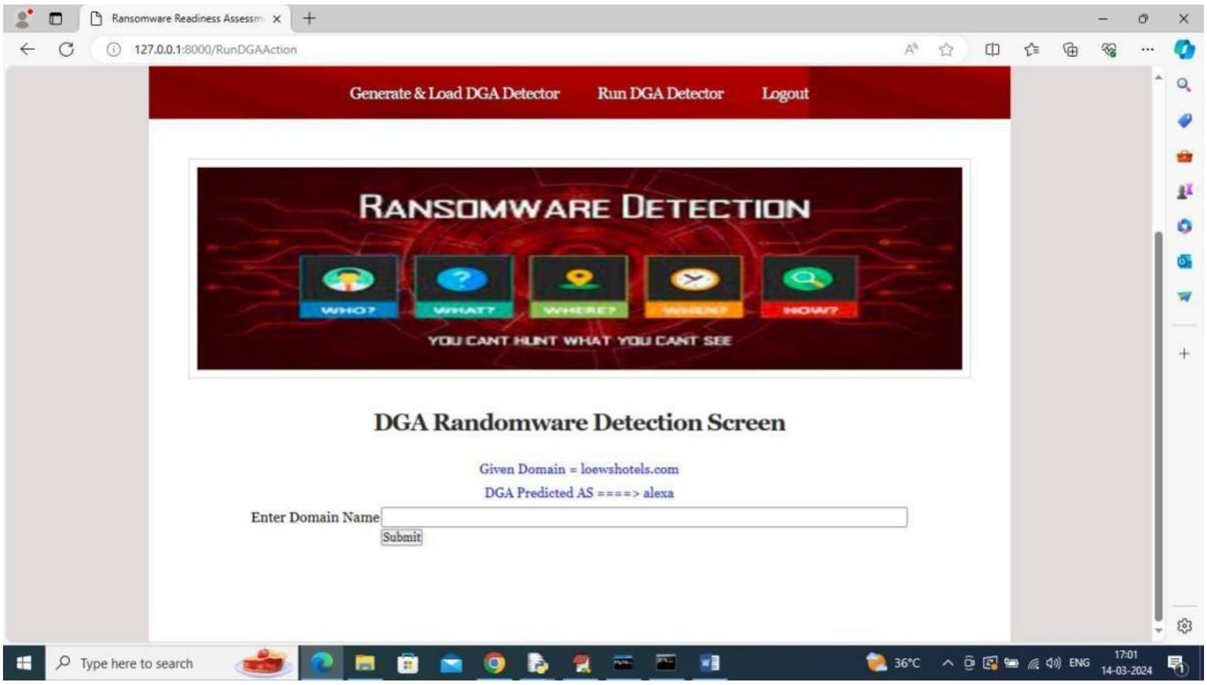


DGA Randomware Detection Screen

Enter Domain Name loewshotels.com

Fig A6: Domain name Entry

Description: In this screen enter some domain name and this domain names you can take from 'Test_domain.txt' file and now click on 'Submit' button to get below page.


Ransomware Readiness Assessm
127.0.0.1:8000/RunDGAAction
Generate & Load DGA Detector
Run DGA Detector
Logout
RANSOMWARE DETECTION
WHO?
WHAT?
WHERE?
WHEN?
HOW?
YOU CANT HUNT WHAT YOU CANT SEE
Type here to search
36°C
ENG
17:01
14-03-2024

DGA Randomware Detection Screen

Given Domain = loewshotels.com

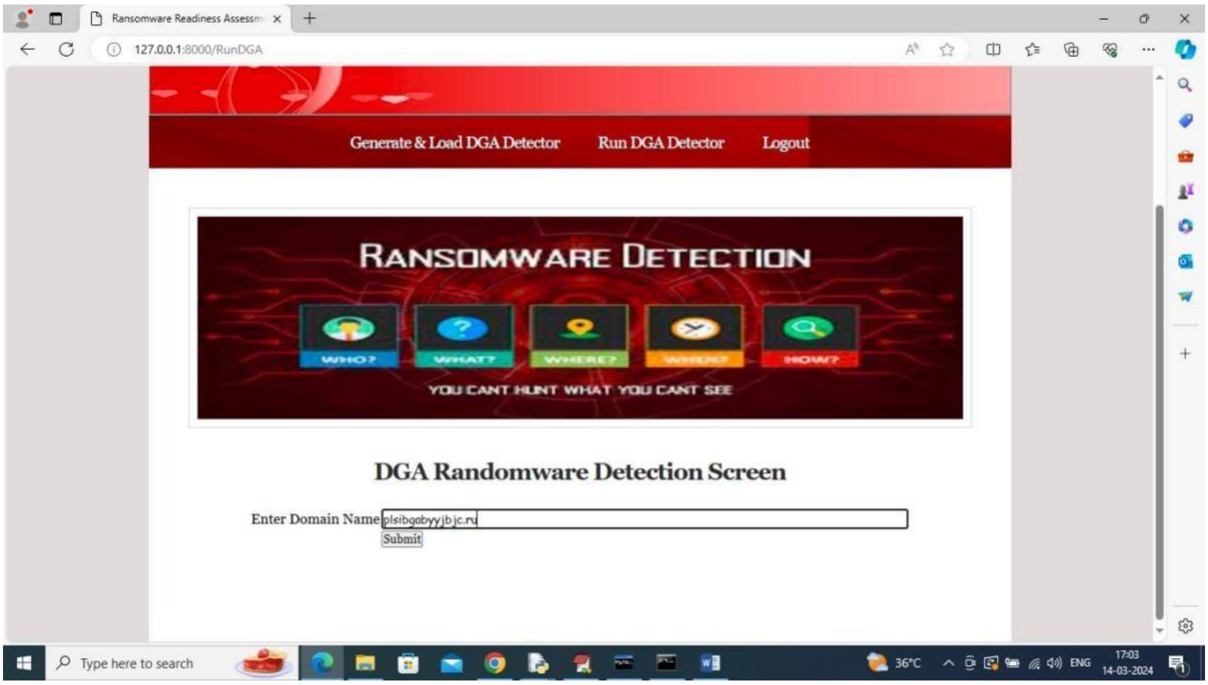
DGA Predicted AS ====> alexa

Enter Domain Name

Submit

Fig A7: DGA Detected as legitimate

Description: In this screen in blue colour text after ==➔ symbol can see domain detected as ‘ALEXA’ and similarly you can enter some domain and get detection result.



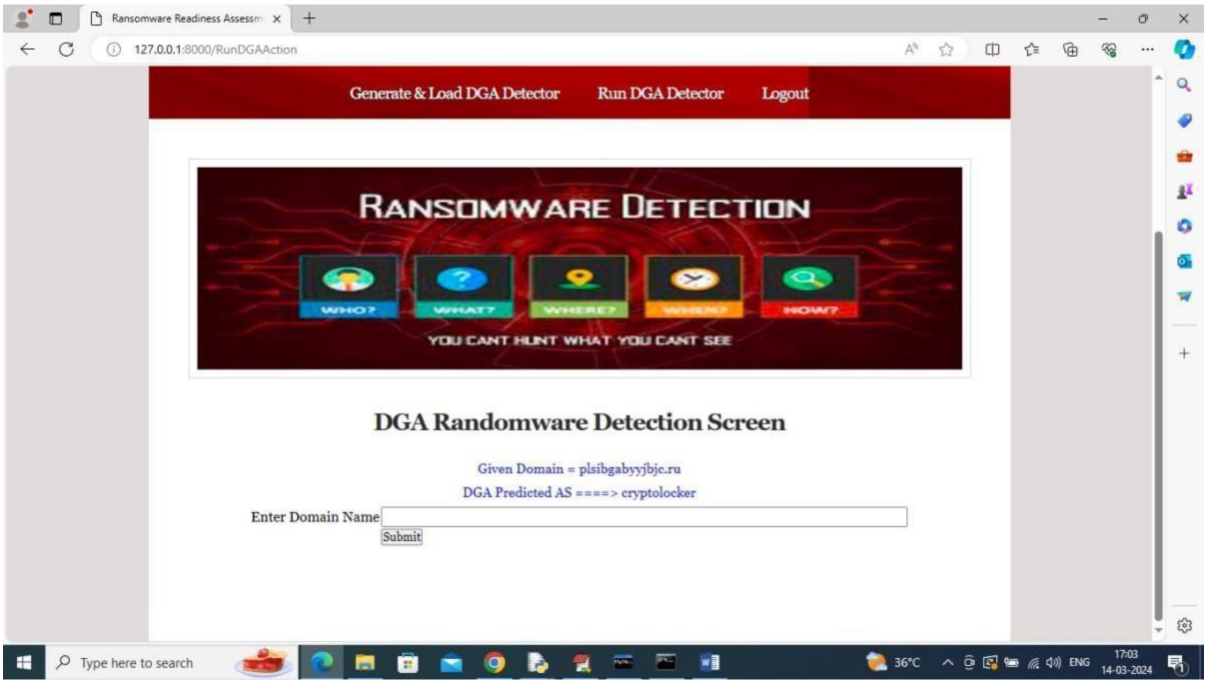
DGA Randomware Detection Screen

Enter Domain Name plsibgobyyjbjc.ru

Submit

Fig A8: Domain name Entry

Description: Enter a domain name for the above-mentioned domain to get below result.


Ransomware Readiness Assessm
127.0.0.1:8000/RunDGAAction
Generate & Load DGA Detector
Run DGA Detector
Logout
RANSOMWARE DETECTION
WHO?
WHAT?
WHERE?
WHEN?
HOW?
YOU CANT HUNT WHAT YOU CANT SEE
Submit
Type here to search
36°C
ENG
17:03
14-03-2024

DGA Randomware Detection Screen

Given Domain = plsibgabyyjbjc.ru

DGA Predicted AS ====> cryptolocker

Enter Domain Name

Fig A9: DGA detected as Ransomware

Description: In this screen domain detected as ‘crypto locker’. Similarly enter any domain and get result.

IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Impact Factor: 7.53

Volume 4, Issue 1, May 2024

Ransomware Readiness Assessment Tool

Dr. Nandini S[1], D P Sai Manohar[2], Darshan D[3], G Shivarame Gowda[4], Nisarga S[5]

Associate Professor, Department of Information Science and Engineering[1]
Under Graduate Student, Department of Information Science and Engineering[2,3,4,5]
S J C Institute of Technology, Chikkaballapur, India

Abstract*:* *Ransomware attacks have been increasingly concerning in times. The situation is only getting worse. They have shed light on a category of software that demands a ransom, for releasing a hostage asset. The majority of ransomware strains rely on encrypting data. Essentially, they lock up files on the victims' devices and network drives before demanding payment to decrypt them. In this study, we first introduce a classification system for ransomware. Then drawing from this taxonomy and identifying a present in highly resilient ransomware during the key exchange process we propose an innovative method for detecting and thwarting these resilient strains to prevent them from encrypting victims' data. Through testing our model shows promising results, in identifying variations of dangerous ransomware strains.*

Keywords: ransomware, crypto virology, prevention; high survivable ransomware

I. INTRODUCTION

Cybercriminals and malware writers have diversified their efforts to make money from their victims, using methods that have been well-established on desktops, laptops, tablets and mobile devices, this includes ransomware. "Ransomware is the name of a so-called phenomenon. It has been built upon the two words ransom and malware" [1]. To define this word, one may give the following general definition: " ransomware is a type of malware which restricts access to the computer system that it infects, and demands a ransom paid to the creator(s) of the malware in order for the restriction to be removed". Some forms of ransomware encrypt files on the system's hard drive (crypto viral extortion, a threat originally envisioned by Adam Young and Moti Yung [2]), while some may simply lock the system and display messages intended to force the user into payment.

Anandrao said in [3] that "It does not appear that a properly designed crypto viral extortion attack has ever beencarried out to date immensely." Also Gazet said in [1] that "No ransomware has reached a sufficient complexity level to successfully become a perfect extortion mean. None of the ransomwares we have studied, presents a reliable perfect extortion scheme. An explanation of this may be that ransomwares' writers have a limited knowledge of cryptography." These statements were valid before 2013. But the Crypto Locker ransomware in 2013 showed that the situation has changed and malware developers have increased their cryptology knowledge.

In June 2013, McAfee released data showing that it had collected over 250,000 unique samples of ransomware in the first quarter of 2013, more than double the number it had obtained in the first quarter of 2012[4]. Crypto Locker surfaced in late-2013, had procured an estimated US$3 million before it was taken down [5]. Based on Bitcoin transaction information ZDNet estimated that the operators of Crypto Locker had procured about US$27 million from infected users [6].

In this paper, we present a novel approach for the most dangerous ransomwares to detect their malicious activity and abort their encryption process before it starts. In summary, wemake the following contributions:

- In the beginning we introduce a classification system for ransomware in section 2 drawing from our research, on attacks and various ransomware types. Our taxonomy aims to encompass all known variants of ransomware
- Moving on to section 3 we outline a method for identifying HSRs that utilize domain generation algorithms (DGA).
- Concluding our discussion we propose a strategy named "Connection Monitor & Connection Breaker" (CM&CB) for combating the potent form of ransomware HSR. Our experimental findings from a proof-of-concept implementation validate the effectiveness of this approach, in mitigating the threat posed by the ransomware strains.

DOI: 10.48175/IJARSCT-18062

II. PROPOSED RANSOMWARE TAXONOMY

In this section we describe our proposed comprehensive taxonomy.

A. Non-Cryptographic Ransomware (NCR)

There are ransomware payloads that don't encrypt. Generally, the payload in these scenarios is only an application that locks the screen or even modifies the partition table and/or master boot record to limit user interaction with the system. This kind of ransomware's weak techniques allow for the restoration of its damages without having to pay the ransom.

B. Cryptographic Ransomware (CGR)

Cryptographic Ransomware (CGR) ensnares precious assets and demands a ransom in exchange for its release via cryptographic techniques. In this context, a common occurrence is that the malware may begin covertly encrypting user data (documents, photos, and so forth). The target user is notified that all of his or her data has been encrypted and that the only way to get it decrypted is to pay the ransom. The cryptosystems that these ransomwares employ allow us to subdivide them into three categories.

III. CONNECTION-MONITOR & CONNECTION-BREAKER APPROACH

It is evident from outlining the taxonomy of ransomware that hybrid cryptosystem ransomwares, or HCRs, pose the greatest threat. In this work, we present CM&CB, a novel framework designed to identify the most perilous varieties of ransomware and stop them from encrypting the files belonging to the affected party. This section outlines our suggested framework after defining the targeted ransomware kinds.

A. High survivable ransomwares (HSR)

The following outlines the conditions that must be met for a mass extortion method to be effective:

- The ransomware should be regarded as compromised and dangerous since it affects consumers' computers.
- The only person who should be able to remove the infection is the ransomware writer. For malware to be able to demand a ransom, it must have a trustworthy way to extract money. A victim will not pay the ransom if she is able to remove the illness on her own [1]. The decryption key should never be kept on the victim's computer for a flawless extortion, as skilled users or virus analyst possessing basic reverse engineering abilities can easily restore the system to a pristine state.

Based on [2] and the three characteristics of perfect extortion, survival is a problem shared by all ransomwares. A ransomware with a "high survivability property" is defined as follows.

Definition 1: A ransomware is considered to have "high survivability" if it is able to keep control over a critical host resource (RC), allowing access only when necessary. If the ransomware is altered or removed, RC becomes permanently inaccessible, and the only way to decrypt data is to use the Command & Control server (C&C) key while the ransom is being paid.

The HCR subtype comprises the highly survivable ransomwares that have been identified in malware databases during the past ten years. In this study, we present a methodology for detection and prevention of high survivable ransomware (HSR). Furthermore, our system is able to identify any ransomware that operates through a key exchange process.

B. Overview of CM&CB approach

Adleman's research has demonstrated that virus identification is an unsolvable issue, and the efficacy of protection systems based on virus detection is doubtful [11]. Young and Yung have demonstrated that, in the event that asymmetric cryptography is robust, reversing the impact of an HCR on the host system may be an unsolvable computing challenge [2]. Our suggested structure can identify every HSR that is now in use (that has been made public so far) prior to the encryption process beginning, hence totally blocking the operation. To understand this feature we review the HCR attack process in more detail.

Step 1 (Find a Victim): The HCR is initially spread through mail spam and other means. For instance, the CryptoLocker is usually distributed by emails sent to business email accounts posing as FedEx, UPS, or DHS customer support-related queries. When the zip attachment in these emails is opened, the PC becomes infected.

Step 2 (doing): In this phase, social engineering techniques are used to have an unsuspecting user carry out the HCR.For instance, the executables included in the CryptoLocker zip files are essentially PDF files masquerading as executables; they often have the extension FORM_101513.pdf.exe and a PDF icon. Because Microsoft does not display extensions by default, when users open them, they appear to be regular PDF files. In some sophisticated HCR like Cryptolocker in thisstep the HCR tries to delete the victim's volume shadow copies,so the restoration will be disabled.

Step 3 (public key exchange): As per our explanation in PuCR, ransomware authors have numerous restrictions when it comes to integrating pair keys into their malware. Consequently, the HCR will endeavour to locate a live C&C or the user's public directory in order to obtain the unique public key Kpu. For instance, Cryptolocker connects to domains produced by a DGA in an effort to locate a live C&C. The DGA will produce domain names such as jkaeaxjmnxvpv.ru and kjqwymybbdrew.biz. It will communicate with a real C&C server once it is found, obtaining a public encryption key that will be used to encrypt data files.

Step 4 (Encryption): Following the acquisition of the infection-specific public key, the victim's data will first be encrypted with Ks and then chained using CBC as the chaining mechanism. After then, the real data can be replaced or erased. Similar to (1), the symmetric key is attached to The Initialization Vector, and the public key of the virus writer is used to encrypt it.

$$M' = E_{Kpu}(\{IV, Ks\}) \qquad (1)$$

Step 5 (Display message): The victim's screen displays the M' and the anonymous ways to get in touch with the HCR writer after infection.

Step 6(Decryption): Deciphering The victim shall send M' to the HCR writer if he consents under the condition that the ransom be paid. After that, HCR writer delivers the pair back to the victim after decrypting it with the matching private key Kpr. In certain instances, the HCR writer use an executable programme for decryption rather than transmitting the {IV, Ks}.

Our first version of the framework was created based on an idea connected to the public key exchange stage in the protocol mentioned above, after we analysed over 40 ransomwares in the recent past, took into account the state of anti-malware technologies, and looked towards the future of malware and anti-malware technologies. Because embedding a static list of C&C candidates into ransomwares presents challenges for cybercriminals should the malicious code finally be captured and examined by security vendors and analysts, the majority of sophisticated and evasive ransomwares at this moment use DGA. Most contemporary ransomware has moved away from hard-coded lists and is built to use DGAs in order to get around this weakness. We created a connection monitor in our initial architecture that could identify DNS domain requests made by DGAs

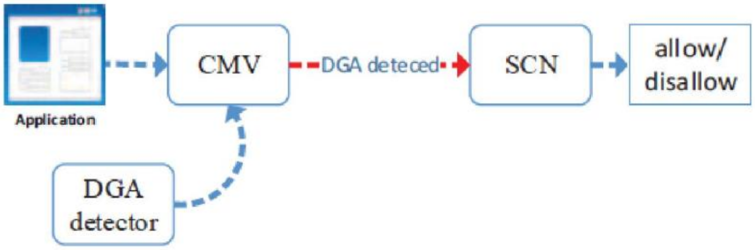


Figure 1: Architecture of DGA detector framework.

As demonstrated in Fig. 1, the connection monitor verifier (CMV) uses DGA detector to identify suspicious connections when ransomware attempts to establish a connection to its C&C via DGA. Another crucial aspect of DGA malicious requests is the speed at which these algorithms originate and request connections from numerous sites. The user can then break the connection and report this suspicious connection address to specialists when the suspicious connection notifier (SCN) alerts them to a strange connection. In addition to all the advantages, the DGA detector structure requires great precision in order to reduce false positives. However, certain domains are not nonsense; occasionally, they are written in a different language. As a result, additional work is required for this framework to truly be accepted.

Since the majority of ransomware targets the Microsoft Windows operating system, we have incorporated a connection-monitor in this framework that checks all applications, particularly the new or untrusted executable files in Windows (Fig. 2). To put it simply, a connection monitor approach looks through all of the executables' outgoing communication and prompts the user to accept or reject the connection. We created an enhanced code signing certificate for our advanced mode of operation. In addition to the standard code signing for integrity checks, we advise developers to submit their certificate authority (CA) the connection addresses needed for their applications. Following a quick addresscheck, the CA will confirm that the list has the verified necessary connection addresses (VRCA).

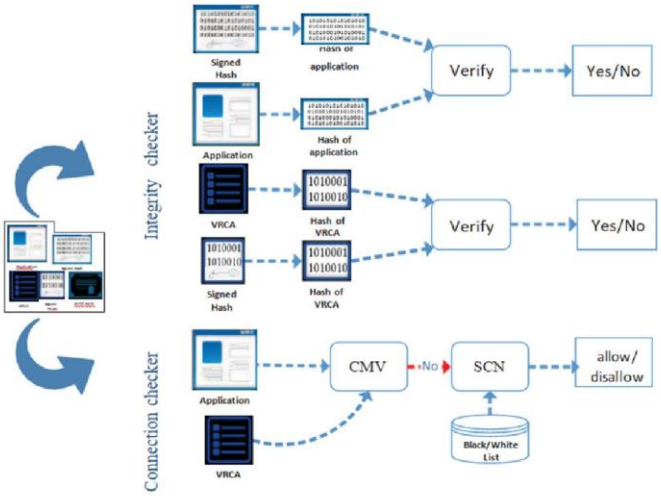


Figure 2: Architecture of proposed framework

A malware analyst can always find the contents of the main HCR body by simply decrypting it using the stored keys. This process will be interrupted if the HCR initially begins encrypting data and then tries to exchange the Kpu with connection breaking. The IV and Ks will be reminded within every instance of the HCR.

IV. EVALUATION

Here, we report the experimental findings and talk about our experiences with this new strategy. Specifically, we evaluate the efficacy of the suggested methodology in identifying and impeding HCRs from using hybrid cryptosystemsto encrypt user data. Some ransomware detectors, such BitDefender AntiCryptoWall, Hitman Pro Kickstart, and HitmanPro CryptoGuard, are signature-based and unable to identify newly discovered or unidentified ransomwares. Wewere unable to compare the results of our framework's detection with those of other tools or frameworks since there is currently no ransomware detector that can identify novel or unidentified ransomwares. We test our concept with over20 new typical ransomware samples to show that our method is capable of recognising HSRs. With the help of this framework, it was possible to identify every HSR and prevent its encryption before the public key exchange was finished. With regard to HSR detection, the suggested method has a 100% detection rate and 0% false negatives. Table 1 gives an overview of various tests. These examples were chosen because they are extremely complex and widely used (from BleepingComputer.com and malwaretips.com). Table 1 defines detection as defeating the encryption.

DOI: 10.48175/IJARSCT-18062

Table 1. PROPOSED FRAMEWORK EXPERIMENTAL RESULTS

Ransomware Name	Ransomware Type				**HSR**	**Detection**
	HCR	***PuCR***	***PrCR***	***NCR***		
Cryptolocker	1	×	×	×	1	1
Cryptolocker 2	1	×	×	×	1	1
Cryptolocker 3	1	×	×	×	1	1
Cryptowall	1	×	×	×	×	1
Cryptowall 2	1	×	×	×	1	1
Cryptowall 3	1	×	×	×	1	1
CoinVault	1	×	×	×	1	1
CryptoGraphic Locker	1	×	×	×	×	1
CryptoDefense	1	×	×	×	×	×
CryptoDefense 2	1	×	×	×	1	1
CryptorBit	×	×	1	×	×	×
TorrentLocker (original)	×	×	1	×	×	×
TorrentLocker	1	×	×	×	1	1
ACCDFISA	×	×	1	×	×	×
BuyUnlockCode	1	×	×	×	×	×
CryptoFortress	1	×	×	×	×	×
PClock2	×	×	1	×	×	×
Critroni(CTB Locker)	1	×	×	×	×	×
Computer Crime & Intellectual Property Section	×	×	×	1	×	×
Harasom	×	×	1	×	×	×

V. CONCLUSION

We discussed a number of methods in this research to mitigate the threat posed by malicious ransomware. To identify and stop harm caused by the most deadly ransomware, new monitoring approaches dubbed CM&CB and a DGA- detector are suggested. The crucial realisation that this strategy needs to succeed is that a key-exchange stage is necessary for HSR operation. The entire HSR process is impeded by monitoring and obstructing this phase. The following succinctly describes the primary benefits of the suggested concept. Initially, this framework is the first of its kind created specifically to address the problem of ransomwares. It does this by keeping an eye on questionable connections and stopping them before they can encrypt the data of the victim. The results of the experimental assessments demonstrate that the suggested framework can effectively block the most potent HSRs, which was previously an unresolved issue in the malware mitigation community. More study is now being conducted on creating a more thorough review. Furthermore, botnets, drive-by download malware, malware that mines bitcoin, and other malicious software can all be detected with the use of this framework. The concept of granting this type of enhanced certificate is not unique to HSRs, hence it can be a helpful defence system against numerous more dangers. Our long- term goal is to expand this framework by incorporating an additional 17 HSR traits in order to identify novel and unidentified advanced HSRs.

REFERENCES

[1] Gazet, Alexandre. "Comparative analysis of various ransomware virii." Journal in computer virology 6.1 (2010): 77-90.

[2] Young, Adam, and Moti Yung. "Cryptovirology: Extortion-based security threats and countermeasures." Security and Privacy, 1996. Proceedings., 1996 IEEE Symposium on. IEEE, 1996.

[3] Shivale, Saurabh Anandrao. "Cryptovirology: Virus Approach." arXiv preprint arXiv:1108.2482 (2011).

[4] "Update: McAfee: Cyber criminals using Android malware and ransomware the most". InfoWorld. Retrieved 16 September 2013.

[5] "Cryptolocker victims to get files back for free". BBC News. 6 August 2014. Retrieved 18 August 2014.

[6] Violet Blue (December 22, 2013). "CryptoLocker's crimewave: A trail of millions in laundered Bitcoin". ZDNet. Retrieved 2013-12-23.

[7] McAfee Threats Report: February 2015, By McAfee Labs,Page 38,2015.

[8] McAfee Threats Report: Third Quarter 2013, By McAfee Labs,Page 19,2013.

[9] McAfee Threats Report: Second Quarter 2014, By McAfee Labs,Page 21 ,2014

[10] Young, Adam, and Moti Yung. Malicious cryptography: Exposing cryptovirology. John Wiley & Sons, 2004.

[11] Adleman, Leonard M. "An abstract theory of computer viruses." Proceedings on Advances in cryptology. Springer-Verlag New York, Inc., 1990

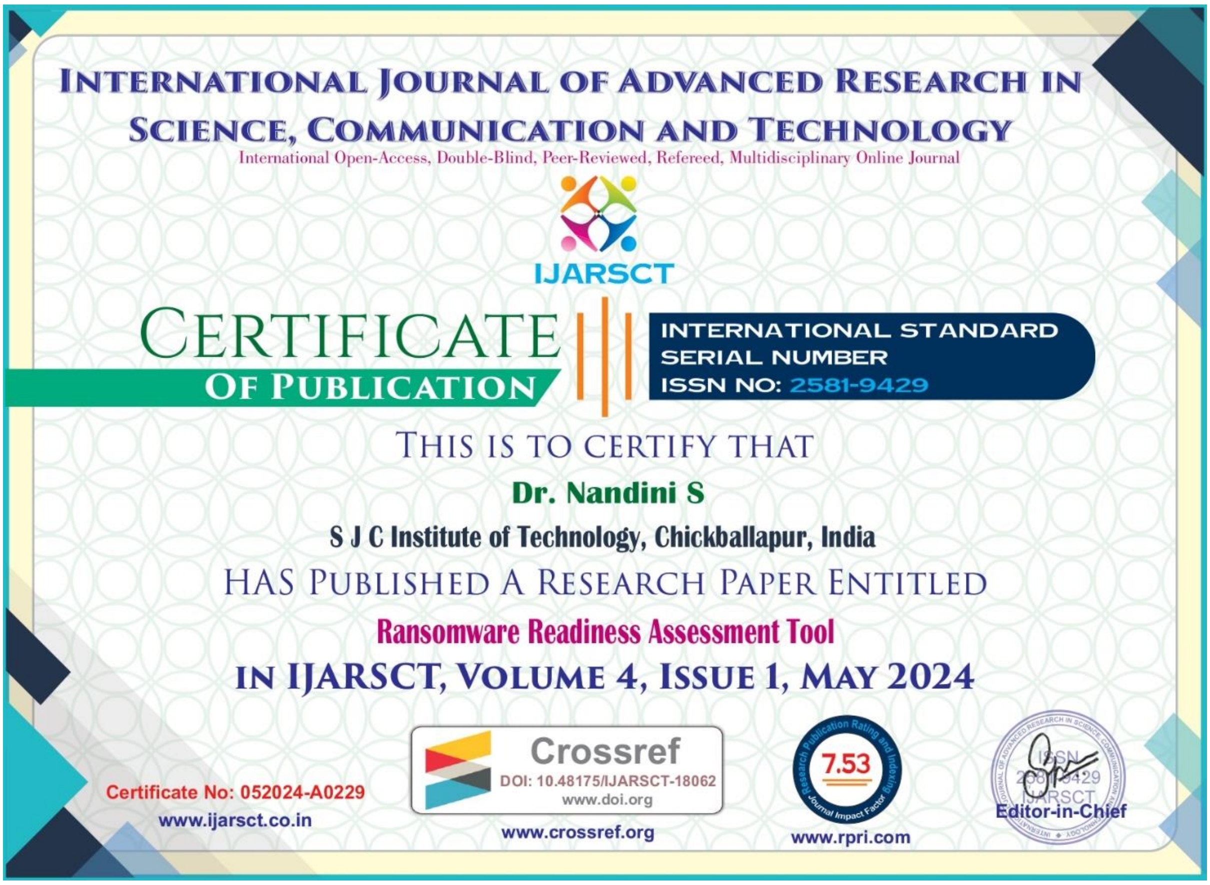
[12] Abidin, Shafiqul, Rajeev Kumar, and Varun Tiwari. "A Review Report on Cryptovirology and Cryptography." International Journal of Scientific & Engineering Research 3.11 (2012): 1.

[13] Stone-Gross, Brett, et al. "Your botnet is my botnet: analysis of a botnet takeover." Proceedings of the 16th ACM conference on Computer and communications security. ACM, 2009.

[14] Yadav, S., Ashwath K. K. R., and Supranamaya R. . "Detecting algorithmically generated domain-flux attacks with DNS traffic analysis." Networking, IEEE/ACM Transactions on 20.5 (2012): 1663- 1677

DOI: 10.48175/IJARSCT-18062

Certificate: Dr. Nandini S



INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN SCIENCE, COMMUNICATION AND TECHNOLOGY

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

IJARSCT

CERTIFICATE OF PUBLICATION

INTERNATIONAL STANDARD SERIAL NUMBER
ISSN NO: 2581-9429

THIS IS TO CERTIFY THAT

Dr. Nandini S

S J C Institute of Technology, Chickballapur, India

HAS PUBLISHED A RESEARCH PAPER ENTITLED

Ransomware Readiness Assessment Tool

IN IJARSCT, VOLUME 4, ISSUE 1, MAY 2024

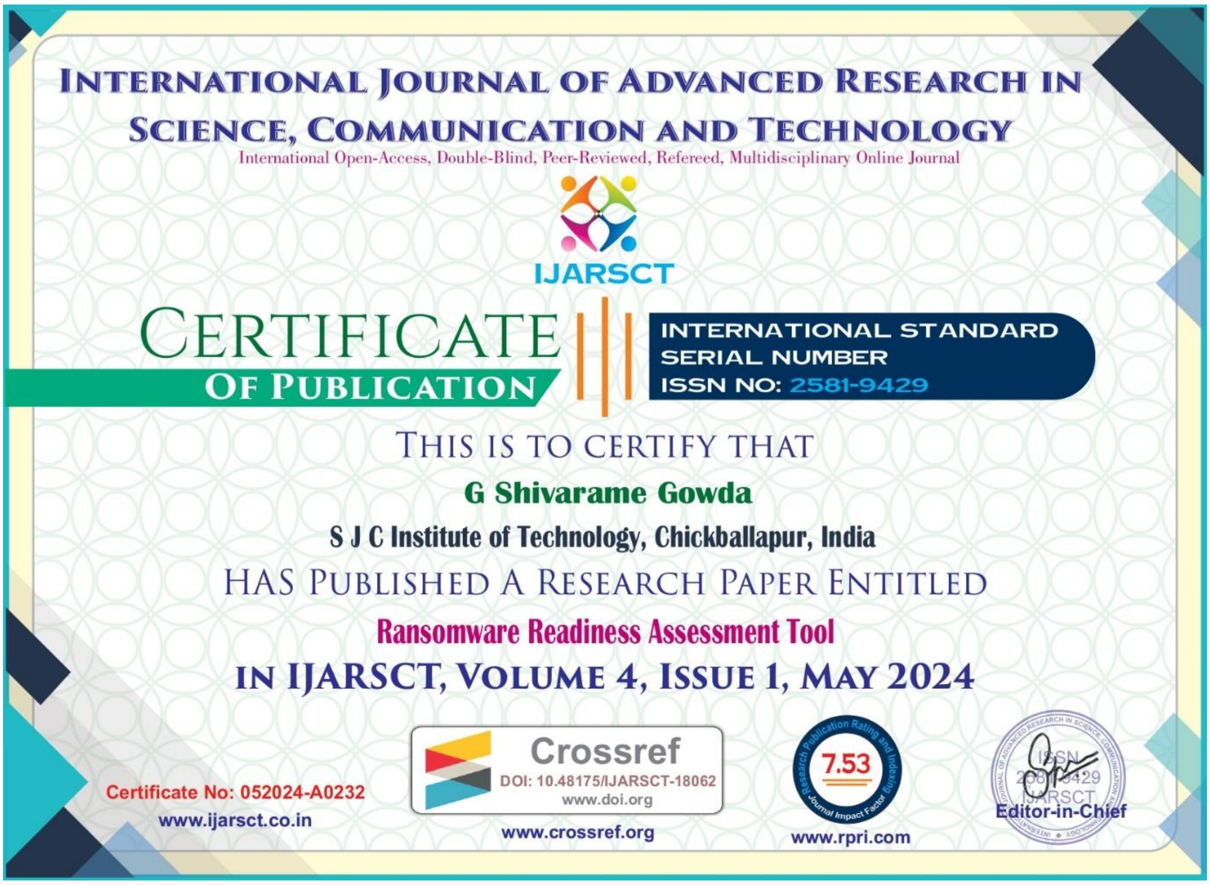
Certificate No: 052024-A0229
www.ijarsct.co.in

Crossref
DOI: 10.48175/IJARSCT-18062
www.doi.org
www.crossref.org

Research Publication Rating and Indexing
7.53
Journal Impact Factor
www.rpri.com

ISSN 2581-9429 IJARSCT
Editor-in-Chief

Certificate: G Shivarame Gowda



INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN SCIENCE, COMMUNICATION AND TECHNOLOGY

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

IJARSCT

CERTIFICATE OF PUBLICATION

INTERNATIONAL STANDARD SERIAL NUMBER
ISSN NO: 2581-9429

THIS IS TO CERTIFY THAT

G Shivarame Gowda

S J C Institute of Technology, Chickballapur, India

HAS PUBLISHED A RESEARCH PAPER ENTITLED

Ransomware Readiness Assessment Tool

IN IJARSCT, VOLUME 4, ISSUE 1, MAY 2024

Certificate No: 052024-A0232
www.ijarsct.co.in

Crossref
DOI: 10.48175/IJARSCT-18062
www.doi.org
www.crossref.org

Research Publication Rating and Indexing
7.53
Journal Impact Factor
www.rpri.com

ISSN 2581-9429 IJARSCT
Editor-in-Chief