

DECRYPTOR & ANALYZER

A PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

MASTER OF SCIENCE (INFORMATION TECHNOLOGY)

BY

SARUNGALE SONIK SANDIP
2510445

UNDER THE GUIDANCE OF
DR. HIREN DAND

COORDINATOR, DEPARTMENT OF INFORMATION TECHNOLOGY



DEPARTMENT OF INFORMATION TECHNOLOGY

PARLE TILAK VIDYALAYA ASSOCIATION'S
MULUND COLLEGE OF COMMERCE

(AFFILIATED TO UNIVERSITY OF MUMBAI)

NAAC RE-ACCREDITED A GRADE – III CYCLE

MULUND WEST, MUMBAI 400080

MAHARASHTRA, INDIA

2021-22

PROJECT PROPOSAL

1.	PRN No	:	2017016400433447
2.	Seat No	:	2510445
3.	Name of the Student	:	Sarungale Sonik Sandip
4.	Title of the Project	:	Decryptor & Analyzer
5.	Name of the Guide	:	Dr. Hiren Dand
6.	Teaching Experience of Guide	:	29 years
7.	Is this your first Submission	:	Yes

Signature of the Student
Guide

Signature of the Coordinator /

Date:

Date:

DECLARATION

I, hereby declare that the project entitled, “**DECRYPTOR & ANALYZER**” done at **Mulund College of Commerce**, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of **MASTER OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

Sarungale Sonik Sandip

ACKNOWLEDGEMENT

The project presented, as part of the curriculum, was the first experience of this kind for me. I had considered this project not only as a program of studies to be completed, but as a goal to learn, study, develop and test commercial software technologies.

I am pleased to be able to say that, in an acceptable manner, I have achieved my goals and goals to make this project a result. I would like to thank and thank the support of some who have helped physically, mentally and intellectually during this project.

Foremost regards to my guide and Co-ordinator **Prof, Dr. Hiren Dand** and **Principal, Dr. Sonali Pednekar** who made available the facilities required for the project work.

I also want to mention the tacit support of my parents who, as always, helped me as much as possible to make my job a success.

The contribution made by my friends and mates, directly or indirectly was indispensable, and will always be remembered.

This opportunity has given me a valuable experience about software development.

ABSTRACT

DECRYPTOR & ANALYZER is a Web based application which will let the user decrypt different cryptographic method in GUI (Graphic user Interface) mode and also analyse different vulnerabilities that might exist in your web application. The developed application is a web application which has two different part's. First is decrypter, which on receiving various encrypted text and type from the user will process and execute them accordingly. Second is Analyzer which basically on receiving url from user will check for some vulnerabilities in your application.

Objectives of the Project:

1. User friendly cryptography.
2. Decrypt text for various cryptographic algorithms.
3. Common vulnerability analyzer.

Languages:

Front-end: HTML, CSS, JavaScript

Back-end: PHP

Database: MYSQL

Table of Content

1. INTRODUCTION:	1
1.1 Background:	1
1.2 Objectives:	2
1.3 Purpose and Scope:	3
1.4 Applicability:	4
1.5 Organization Report	5
2. SURVEY OF TECHNOLOGIES	7
2.1 Technologies available for development	7
2.2 Tools and Techniques used	7
2.3 Programming Languages, framework and library used for development	8
2.4 Major Website Programming Languages and Framework Available For Development	11
2.5 Algorithm performance	12
2.6 Similar Technology/Application	21
2.7 Testing Technology	37
3. REQUIREMENTS AND ANALYSIS	38
3.1 Problem Defination:	38
3.2 Requirements Specification:	39
3.3 Project Schedule:	42
3.4 Software and Hardware Requirements	42
3.5 Conceptual models	43
4. SYSTEM DESIGN	47
4.1 Basic Modules:	47
4.2 Data Design:	48
4.3 Procedural Design:	50
4.4 User Interface Design:	54
4.5 Security Issues:	56
4.6 Test Cases Design:	58
CHAPTER 5:-IMPLEMENTATION AND TESTING	60

5.1:-Implementation Approach:-	60
5.2:-Coding Details & Code Efficiency :-	61
5.3:-Testing approach:-	78
5.4:-Modification and Improvements	83
CHAPTER 6:-RESULTS AND DISCUSSION	84
6.1:-Test Reports:-	84
6.2:- User Documentation	85
CHAPTER 7:-CONCLUSIONS	87
7.1:- Significance of the system	87
7.2:- Limitations of the system	87
7.3:- Future scope of the project	87

1. INTRODUCTION:

1.1 Background:

Data encryption is used all over the place in today's connected society. As a modern society becomes more connected, and more information becomes available there is need for safeguards which bring data integrity and data secrecy. In addition, authenticating the source of information gives the recipient, with complete certainty that the information came from the original source and that it has not been altered from its original state.

Data encryption translates data into another form, or code, so that only people with access to a secret key formally called a decryption key or password can read it. Encrypted data which is cannot be read by human is called cipher text, while human readable data is called plain text. Currently, encryption is one of the most popular and effective data security methods used by organizations. Types of data encryption are divided into two types which are symmetric encryption and asymmetric encryption. The purpose of data encryption is to protect digital data confidentiality as it is stored on computer systems and transmitted using the internet or other computer networks. The examples of encryption algorithm that popular are SHA256, MD5 and many more. Usually this algorithms use one way encryption means it can not be decrypted. But there are some methods using which you can try to decrypt this encrypted data.

A web based application might have various injection vulnerabilities like SQL Injection, NoSQL Injection, XSS & CSRF. Which can be avoided using proper development practices. But sometime this vulnerabilities can still exist in application despite following proper guideline. As web application can have many input fields, developer might miss this guideline in some of your application and this can lead to security breach.

This project mainly concentrates about developing a Web based application which will let the user decrypt different cryptographic method in GUI (Graphic user Interface) mode and also analyse different vulnerabilities that might exist in your web application. The developed application is a web application which has two different part's. First is decrypter, which on receiving various encrypted text and type from the user will process and execute

them accordingly. Second is Analyzer which basically on receiving url from user will check for some vulnerabilities in your application.

1.2 Objectives:

- **Encrypt your text in various cryptographic algorithms.**

Basic aim is to develop a user friendly application so that user can encrypt their text in various cryptographic algorithms.

- **Decrypt text for various cryptographic algorithms.**

Some encrypted text can not be decrypted and some can be only decrypted using private key. This application will try to decrypt using some techniques.

- **User friendly cryptography.**

Cryptographic libraries are based on command line tools and are difficult to be used. It requires sequential instruction to be provided manually through DOS. This application will make this easy through interface.

- **Common vulnerability analyzer**

Analyzer which will perform scan on provided web application and find for some common web application vulnerabilities like CORS and ClickJack.

1.3 Purpose and Scope:

1.3.1 Purpose:

Scope of this project is to create a application where user can analyze and find their web application vulnerabilities. Also can encrypt their text in various encryption algorithm and can also try to decrypt text for various algorithm using different techniques.

A vulnerability analyzer will be usefull to find loopholes in application which developer might have missed. Find vulnerabilities in your application is important before others find it.

Decrypter can be use to decrypt text which has one way encryption method or private key. Decrypter use various techniques to try to decrypt text.

1.3.2 Scope:

In this project, we deal with the concepts of cryptography and vulnerability analyzer. This project is designed of combining the vulnerability scanner and cryptography features factors.

This project is based on PHP but sometimes ASP or Python is used instead of PHP for some algorithms using API.

1.4 Applicability:

- When the user opens the website he will see two options : Decrypter & Analyzer.
- After selecting Decrypter, The next screen will appear where he can select algorithm, decryption type, will enter his text and start decryption process.
- Decryption type can be Rainbow table, key, brute force - Numeric brute force & Alpha numeric brute force, Dictionary attack, Common words, World list & customize world list.
- User can also use encrypter to encrypt their text in available algorithms. User just needs to select algorithm and enter text and click on encrypt button.
- After selecting Analyzer from dashboard, The next screen will appear where user will enter his URL and select attack type and start scan. Type of attack include.
 1. SQL Injection
 - Here system will try to inject sql injection in input fields and try to find Vulnerabilities.
 - System will try to entry various sql injection and depending on output it depending on output it will determine success or failure.
 2. XSS
 - In this attack system will try to execute various xss scripts depending on output it depending on output it will determine success or failure.
 3. Broken authentication
 - System will visit other pages in application and try to check if any page is accessible without authentication.

1.5 Organization Report

Chapter 1: Introduction

The outline has several parts as given below:

Background: An explanation of the development context and its association with the work already done in the field.

Objectives: Concise declaration of the goals and aims of the plan.

Purpose: Project theme description that answers questions about why this project is implemented.

Scope: A ephemeral summary of the methodology, hypotheses and limits.

Applicability: The student must describe the direct and indirect applications of his work.

Achievements: Explain what knowledge the student achieved after the accomplishment of the work.

Chapter 2: Survey of Technologies:

The Technology Survey demonstrates the acquaintance and understanding of available technology associated to the project. Provide facts of all related technologies desired to accomplish the project goal. Describe the technologies obtainable in the chosen area and present a reasonable study of all available technologies and which is the best of them and why choose the technology.

Chapter 3: Requirements and Analysis:

This chapter defines what problem you are experiencing and how your project will overcome the problem that is occurring in society. Listing the necessities of the project. Scheduling and planning for the project so that project should complete on time and doesn't go in critical path. Give the list of software along with hardware components require by project and explain key points about it.

Chapter 4: System Design:

This chapter describes features and procedures researched in aspects, consisting of screen layouts, business rules, process diagrams, pseudo-code, and other documentation, which is appropriate for the project. It consists of designing a system, i.e. designing schemes, designing algorithms and designing basic modules of a system.

Chapter 5: Implementation and Testing

This chapter contains the details of various modules, codes and various libraries I have imported successfully to implement the project. It also contains the details of process or work I have done to make code efficient.

This chapter also contains the details of various testing approach I have used.

Chapter 6: Results and Discussion

This Chapter shows the details of the test results after testing the software and generate report on the basis of the Test results.

This chapter will also show the behaviour of the Website when inputs are different from the one written in the Test Cases.

This Chapter also contains details of working of the website and the different functions in the project. It should also contain User Manual, Which provides the understanding of the working of the project to the user.

Chapter 7: Conclusion

This chapter summaries about all the important point of all other chapters.

It also shows the limitation in the proposed system and details about future plan to explore the scope of the project.

2. SURVEY OF TECHNOLOGIES

2.1 Technologies available for development

Website development sector is grown in recent years and it is in boom in the market. There are several technologies for web development widely used. In India, Web Sites can have numerous capacities and can be utilized in different style. Website development tools and technologies refer to a variety of tools and programming languages used to perform the various activities involved in developing Web applications.

2.2 Tools and Techniques used

1. Sublime Text

- Sublime Text is a light weight cross-platform source code editor. It supports many programming languages and markup languages, functions can be added by the users with a plugin. Sublime Text is available for Windows, Mac and Linux.

2. Adobe XD

- Adobe XD is use to design some component of the application which will help to give nice look and feel.

3. Apache

- Apache is a web server that powers around 46% of web sites round the internet. It is maintained and developed by Apache Software Foundation.
- Apache is a web server but it is not a physical server, it is a software that runs on a server. Its job is to establish a connection between a server and browser like Firefox, Google Chrome, Safari, etc. While transferring files back and forth between the client and server. Apache is a cross-platform software, it works on both UNIX & Window.

4. WAMP

- WAMP Server is a software stack for the Windows operating system, created by Romain Bourdon and has Apache web server, Open SSL for SSL support, MySQL database and PHP programming language.
- Full form of WAMP is
 - W – Window
 - A – Apache
 - M – MYSQL
 - P - PHP

5. Browser

- A Browser is alternatively referred as a web browser or an Internet browser. It is a software used for exploring and presenting the contents on the World Wide Web. There are many browsers e.g. Google Chrome, Safari, Mozilla Firefox, Opera etc. To see the actual output of the website the browsers are used. The languages like HTML, CSS and JS are compiled by the browser.

2.3 Programming Languages, framework and library used for development

1. HTML5:

- Hypertext Markup Language is widely known as HTML.
- HTML is used to make structure of a webpage.
- HTML define structure and layout of webpage by using tags and attributes.
- It is widely used language on web for website development.
- Currently HTML5 is been mostly used in market because of its new features and wide support of tags.
- Many old tags like blink tag and marquee tag is been removed in HTML5.

2. CSS3:

- Cascading Style Sheet (CSS) is a style sheet language used to design the webpage.
- The CSS is used along with HTML.
- HTML describes the content of the website, while CSS tells how the content should be placed.
- CSS provide an attractive look to a webpage.
- The newer version of CSS is CSS3.
- It support responsiveness to the webpage and other features like:
 - CSS Animations and Transitions
 - Calculating Values With calc()
 - Advanced Selectors
 - Generated Content and Counters
 - Gradients
 - Webfonts
 - Box Sizing
 - Border Images
 - Media Queries
 - Multiple Backgrounds
 - CSS Columns
 - CSS 3D Transforms

3. JavaScript:

- JavaScript is client-side scripting language.
- JavaScript use to make interactive web pages and it is an important part of web development.
- JavaScript is not a programming language, it is a scripting language.
- It is developed by Brendan Eich. at Netscape, for the Netscape Navigator Web browser.
- Currently it is one the most popular language as it can be not be used for website frontend but also it can be used for website backend using node JS,

Native android app using React Native, desktop GUI applications using electron JS, Machine learning using Tensor flow JS.

4. PHP:

- PHP is a server side scripting language designed for web development.
- Originally PHP stands for Personal Home Page, but it now PHP stands for PHP: Hypertext Preprocessor.
- PHP code can be executed, embedded in HTML code or by making a separate '.php' file.
- PHP code is run on the server and its output is been send to web browser.
- Browser never interact with PHP code, after executing php on server only output is given to browser to run.
- PHP is safe because user can't see PHP code which is been executed on the server.
- But also there are many security issue with PHP because of bad code written by web developers.
- Despite being old language, PHP is still in top 10 and on of the most demanded language because 78.9% of websites on the internet still use PHP for their website backend.

5. JQuery:

- JQuery is a JavaScript library designed to simplify work.
- JQuery can do things in 1 line of code for which vanilla java script require 3-4 lines.
- In JQuery various things link DOM manipulation, CSS animation, event handling and Ajax can be done easily.
- In 10 million most popular websites 73% website use JQuery.
- JQuery is easy to learn and easy and its syntax is easy to adapt.

6. MYSQL:

- MySQL is a open-source relational database management system.
- MYSQL is used to store data.
- MYSQL uses table's to store data.
- MYSQL is one of the most used database for storing data from websites.
- MYSQL is owned and maintained by Oracle Corporation.

6. ASP dot Net:

- ASP.NET is an open-source, server-side web-application framework designed for web development to produce dynamic web pages.
- It was developed by Microsoft to allow programmers to build dynamic web sites, applications and services.

2.4 Major Website Programming Languages and Framework Available For Development

○ Laravel

Laravel is a PHP based framework. Laravel preferred and acknowledged by many web developers because of its use of MVC concept, neat & clear code and proper folder / file structures.

○ Angular & Angular JS

Angular a JavaScript framework created and backed by Google. Angular is a complete framework based on type script (superset of ES6) and angular JS is a library based on java script.

○ React JS

React JS is a JavaScript library created and backed by Facebook. Currently many website including Facebook use react JS for interacting with user.

○ Vue JS

Vue JS is JavaScript framework. It is like a combination of angular and react JS. It uses some features of react JS and some features of angular.

○ Django

Django is a python based framework. It allow user to write code in python for website backend.

- Ruby on Rail

Ruby on Rail is a framework written in ruby. Rail is a model view controller (MVC) framework.

2.5 Algorithm performance

• MD5



MD5 Encryption

Number of words: 100000
Time(In milliseconds) took to encrypt 100000 by md5: 62

The MD5 message-digest algorithm is a cryptographically broken but still widely used hash function producing a 128-bit hash value. Although MD5 was initially designed to be used as a cryptographic hash function, it has been found to suffer from extensive vulnerabilities. It can still be used as a checksum to verify data integrity, but only against unintentional corruption. It remains suitable for other non-cryptographic purposes, for example for determining the partition for a particular key in a partitioned database, and may be preferred due to lower computational requirements than more recent Secure Hash Algorithms algorithms.

MD5 was designed by Ronald Rivest in 1991 to replace an earlier hash function MD4, and was specified in 1992 as RFC 1321.

One basic requirement of any cryptographic hash function is that it should be computationally infeasible to find two distinct messages that hash to the same value. MD5 fails this requirement catastrophically; such collisions can be found in seconds on an ordinary home computer.

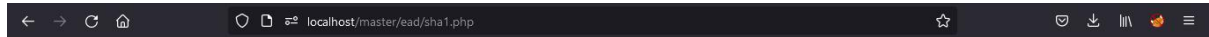
On 31 December 2008, the CMU Software Engineering Institute concluded that MD5 was essentially "cryptographically broken and unsuitable for further use". The weaknesses of MD5 have been exploited in the field, most infamously by the Flame malware in 2012. As of 2019, MD5 continues to be widely used, despite its well-documented weaknesses and deprecation by security experts.

MD5 Algorithms are useful because it is easier to compare and store these smaller hashes than store a large variable length text. It is a widely used algorithm for one-way hashes used to verify without necessarily giving the original value. Unix systems use the MD5 Algorithm to store the passwords of the user in a 128-bit encrypted format. MD5 algorithms are widely used to check the integrity of the files.

- Moreover, it is very easy to generate a message digest of the original message using this algorithm. It can perform the message digest of a message having any number of bits; it is not limited to a message in the multiples of 8, unlike MD5sum, which is limited to octets.

But for many years, MD5 has prone to hash collision weakness, i.e. it is possible to create the same hash function for two different inputs. MD5 provides no security over these collision attacks. Instead of MD5, SHA (Secure Hash Algorithm, which produces 160-bit message digest and designed by NSA to be a part of digital signature algorithm) is now acceptable in the cryptographic field for generating the hash function as it is not easy to produce SHA-I collision and till now no collision has been produced yet.

- sha1



SHA1 Encryption

Number of words: 100000

Time(In milliseconds) took to encrypt 100000 by sha1: 53

SHA-1 or Secure Hash Algorithm 1 is a cryptographic hash function which takes an input and produces a 160-bit (20-byte) hash value. This hash value is known as a message digest. This message digest is usually then rendered as a hexadecimal number which is 40 digits long. It is a U.S. Federal Information Processing Standard and was designed by the United States National Security Agency.

In cryptography, SHA-1 (Secure Hash Algorithm 1) is a cryptographically broken but still widely used hash function which takes an input and produces a 160-bit (20-byte) hash value known as a message digest – typically rendered as a hexadecimal number, 40 digits long. It was designed by the United States National Security Agency, and is a U.S. Federal Information Processing Standard.

Since 2005, SHA-1 has not been considered secure against well-funded opponents; as of 2010 many organizations have recommended its replacement. NIST formally deprecated use of SHA-1 in 2011 and disallowed its use for digital signatures in 2013. As of 2020, chosen-prefix attacks against SHA-1 are practical. As such, it is recommended to remove

SHA-1 from products as soon as possible and instead use SHA-2 or SHA-3. Replacing SHA-1 is urgent where it is used for digital signatures.

All major web browser vendors ceased acceptance of SHA-1 SSL certificates in 2017. In February 2017, CWI Amsterdam and Google announced they had performed a collision attack against SHA-1, publishing two dissimilar PDF files which produced the same SHA-1 hash. However, SHA-1 is still secure for HMAC.

Microsoft has discontinued SHA-1 code signing support for Windows Update on August 7, 2020.

- **sha256**



SHA256 Encryption

Number of words: 100000

Time(In milliseconds) took to encrypt 100000 by sha256: 110

Among the many advancements seen in network security, encryption and hashing have been the core principles of additional security modules. The secure hash algorithm with a digest size of 256 bits, or the SHA 256 algorithm, is one of the most widely used hash algorithms. While there are other variants, SHA 256 has been at the forefront of real-world applications.

To understand the working of the SHA 256 algorithm, you need first to understand hashing and its functional characteristics.

SHA 256 is a part of the SHA 2 family of algorithms, where SHA stands for Secure Hash Algorithm. Published in 2001, it was a joint effort between the NSA and NIST to introduce a successor to the SHA 1 family, which was slowly losing strength against brute force attacks. The significance of the 256 in the name stands for the final hash digest value, i.e. irrespective of the size of plaintext/cleartext, the hash value will always be 256 bits.

The other algorithms in the SHA family are more or less similar to SHA 256. Now, look into knowing a little more about their guidelines.

Some of the standout features of the SHA algorithm are as follows:

Message Length: The length of the cleartext should be less than 264 bits. The size needs to be in the comparison area to keep the digest as random as possible.

Digest Length: The length of the hash digest should be 256 bits in SHA 256 algorithm, 512 bits in SHA-512, and so on. Bigger digests usually suggest significantly more calculations at the cost of speed and space.

Irreversible: By design, all hash functions such as the SHA 256 are irreversible. You should neither get a plaintext when you have the digest beforehand nor should the digest provide its original value when you pass it through the hash function again.

Now that you got a fair idea about the technical requirements for SHA, you can get into its complete procedure, in the next section.

• SHA512



SHA512 Encryption

Number of words: 100000
Time(In milliseconds) took to encrypt 100000 by sha512: 186

SHA-512, or Secure Hash Algorithm 512, is a hashing algorithm used to convert text of any length into a fixed-size string. Each output produces a SHA-512 length of 512 bits (64 bytes).

This algorithm is commonly used for email addresses hashing, password hashing, and digital record verification. SHA-512 is also used in blockchain technology, with the most notable example being the BitShares network

SHA-512 is just one of several algorithms in the Secure Hashing Algorithm (SHA) family. In 2001, SHA-512 was published by the National Institute of Standards and Technology (NIST) as a U.S. Federal Information Processing Standard (FIPS). Before we look at the specifics of how SHA-512 is used today, let's briefly cover the history of these algorithms.

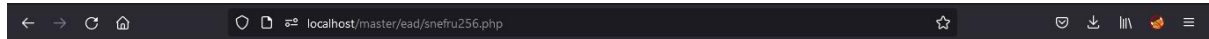
Compared to the SHA-256 algorithm, the adoption of the SHA 512 algorithm by blockchain projects has been very minimal. Most blockchain networks that chose not to implement SHA-256 opted for other hashing algorithms such as Scrypt, Lyra2REv2, Equihash, and CryptoNight. With that being said, here are a few examples of networks that use or have used SHA-512. Overall.

SHA-512 is also used in a variety of non-blockchain applications. It's oftentimes used in conjunction with SHA-256 but sometimes used by itself. Nonetheless, as with blockchain applications, SHA-512 adoption for other technical applications clearly pales in comparison to SHA-256.

SHA-512 was used to authenticate archival video from the International Criminal Tribunal of the Rwandan genocide. Unix and Linux vendors use both SHA-256 and SHA-512 for secure password hashing. An email suppression list solution called OPTIZMO provides the storage and distribution of SHA-512 hashed email addresses for major clients such as Salesforce, LendingTree, Hotwire, and eharmony.

SHA-512 hasn't been able to gain the same level of popularity as SHA-256 or even other types of newer hashing algorithms when it comes to real-world use in blockchain. That being said it does have a few non-blockchain applications that are noteworthy.

- **snefru256**



SNEFRU256 Encryption

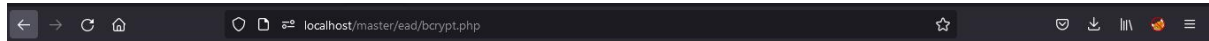
Number of words: 100000

Time(In milliseconds) took to encrypt 100000 by snefru256: 653

Snefru is a cryptographic hash function invented by Ralph Merkle in 1990 while working at Xerox PARC. The function supports 128-bit and 256-bit output. It was named after the Egyptian Pharaoh Sneferu, continuing the tradition of the Khufu and Khafre block ciphers.

The original design of Snefru was shown to be insecure by Eli Biham and Adi Shamir who were able to use differential cryptanalysis to find hash collisions. The design was then modified by increasing the number of iterations of the main pass of the algorithm from two to eight. Although differential cryptanalysis can break the revised version with less complexity than brute force search (a certification weakness), the attack requires $2^{88.5}$ operations and is thus not currently feasible in practice.

• Bcrypt



Bcrypt Encryption

Number of words: 10

Time(In milliseconds) took to encrypt 10 by Bcrypt: 1612

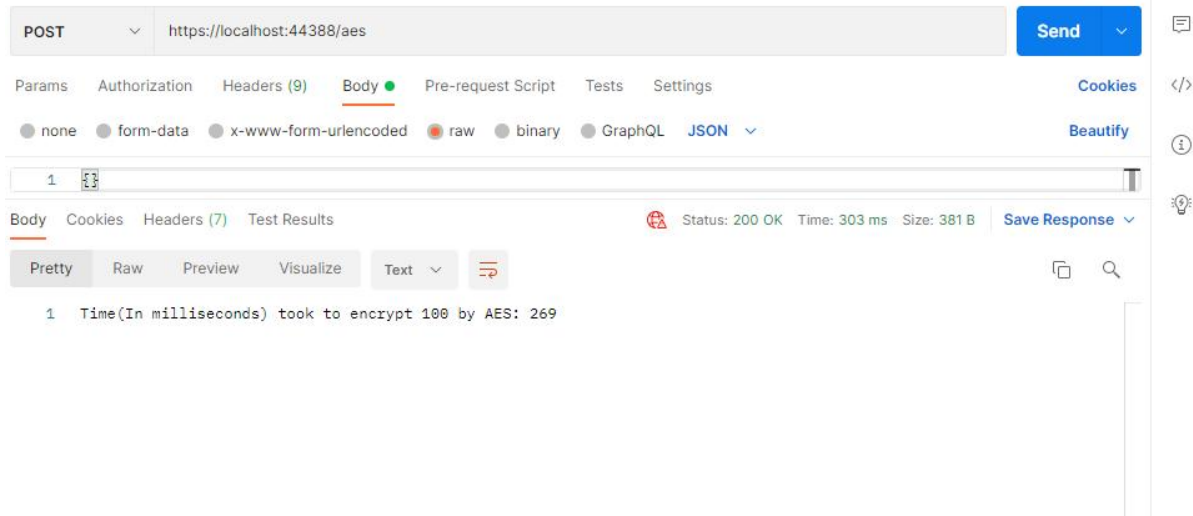
Bcrypt is a password hashing function designed by Niels Provos and David Mazières, based on the Blowfish cipher, and presented at USENIX in 1999.

Bcrypt is a cross platform file encryption utility. Encrypted files are portable across all supported operating systems and processors. Passphrases must be between 8 and 56 characters and are hashed internally to a 448 bit key. However, all characters supplied are significant. The stronger your passphrase, the more secure your data.

In addition to encrypting your data, bcrypt will by default overwrite the original input file with random garbage three times before deleting it in order to thwart data recovery attempts by persons who may gain access to your computer. If you're not quite ready for this level of paranoia yet, see the installation instructions below for how to disable this feature. If you don't think this is paranoid enough.. see below.

Bcrypt uses the blowfish encryption algorithm published by Bruce Schneier in 1993.

• AES



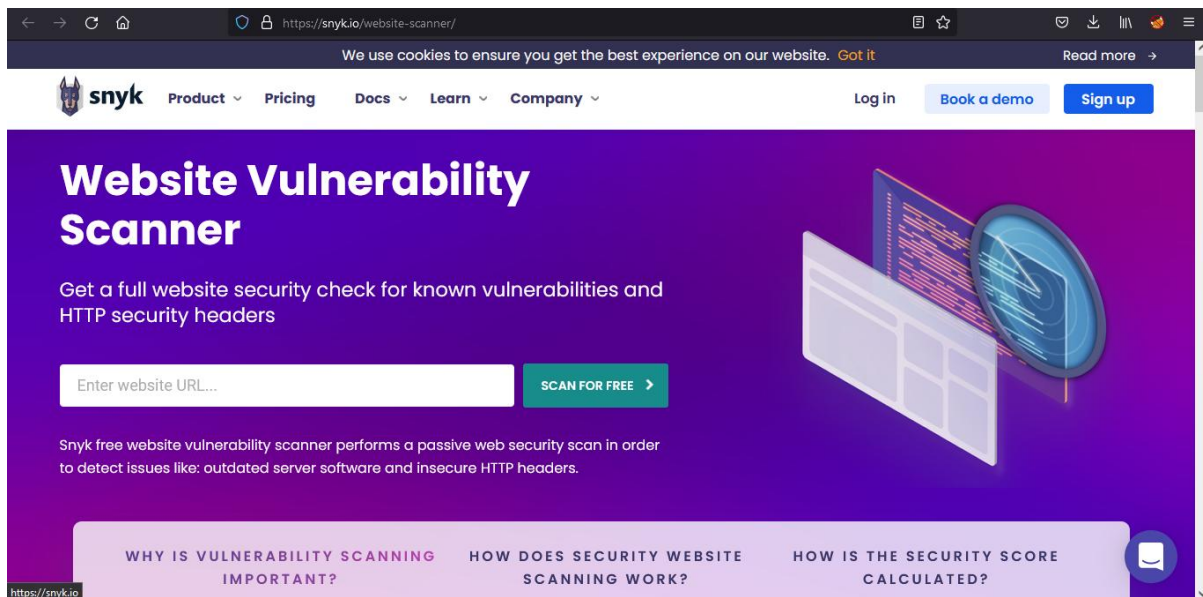
The Advanced Encryption Standard is everywhere, and you don't even know it. When you fill out forms online for the government (passport applications, drivers license renewals, etc.), when you store personal information on a website (Facebook, Twitter, etc.), and even when you use your VISA or bank card to make a purchase, it is in the background, doing its thing.

Why? Because in today's day and age, security is paramount, and protecting information is a significant part of security. Advanced Encryption Standard (AES) is a cipher, meaning that it is a method or process used to change raw information (usually human readable) into something that cannot be read. This part of the process is known as encryption. The method uses a known, external piece of information, called a key, to uniquely change the data. An example might be your computer login password, or the password to your account on a bank machine. Further, the process is reversible, meaning that it can be applied again to put the information back into its original form. This part is known as decryption.

The AES has three fixed 128-bit block ciphers with cryptographic key sizes of 128, 192 and 256 bits. Key size is unlimited, whereas the block size maximum is 256 bits. The AES design is based on a substitution-permutation network (SPN) and does not use the Data Encryption Standard (DES) Feistel network.

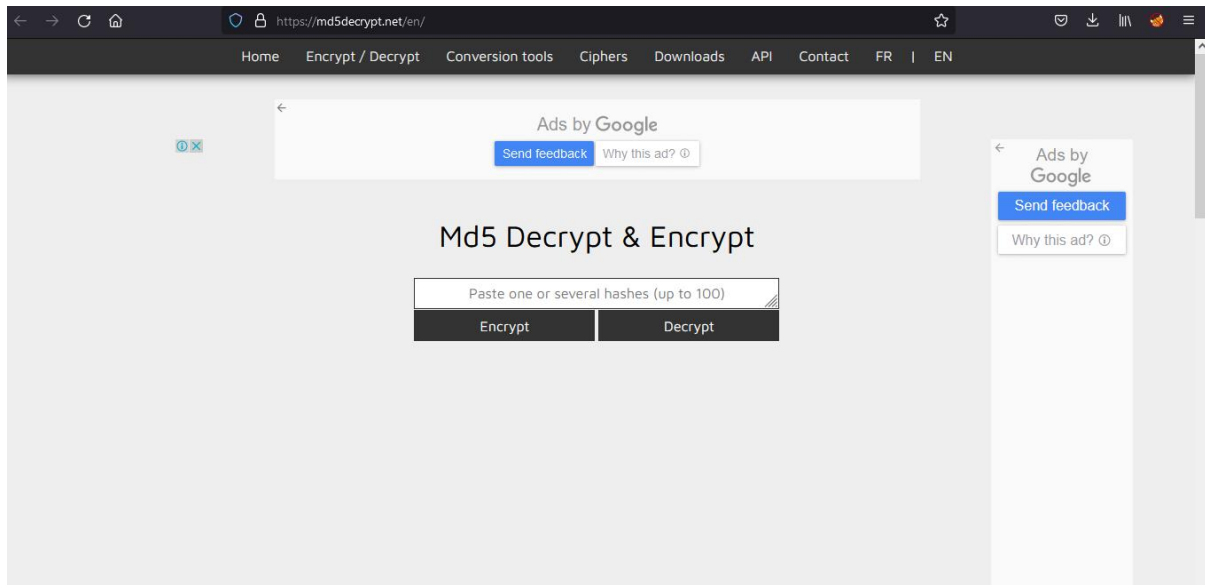
2.6 Similar Technology/Application

Snyk.io:



Vulnerability scanner monitors for misconfigurations or vulnerable third-party open-source dependencies that pose cybersecurity threats. Online vulnerability scanners either rely on a database of known vulnerabilities or probe for common flaw types to discover unknown vulnerabilities. Website scanner logs detect vulnerabilities and assigns a risk score.

Md5decrypt.net:



Md5 (*Message Digest 5*) is a cryptographic function that allows you to make a 128-bits (32 characters) "hash" from any string taken as input, no matter the length (up to 2^{64} bits). This function is irreversible, you can't obtain the plaintext only from the hash. The only way to online decrypt your hash is to compare it with a database using our online decrypter. Here we have 15183605161 md5 online database to help you with decryption. One should know that md5, although it's very used and common, shouldn't be use to encrypt critical data, since it's not secure anymore (collisions were found, and decrypt is becoming more and more easy). If you are building a new website, Sha256, 512, or other kinds of encryption (with salt) would be better than md5 encryption, or even sha1. Our decrypter online database is coming from all the wordlist I was able to find on the internet. I then sorted them, and enlarge the final wordlist by creating a script that multiplied the list to finally lend to a unique and pertinent online md5 hashes list.

2.7 Survey report (2015-2021)

With the increase in internet era there is also an increase in security breaches. There can be multiple reasons for cyber-attacks, but the biggest attacks happened in last 5 years due to poor security lets discuss that.

1. Yahoo! Data Breach

Year of breach: 2013 - 2016

Data breached: 3 billion user accounts

According to data breach statistics, the largest data breach in history is the one that Yahoo! suffered for several years. Not only is it the biggest breach according to the number of affected users, but it also feels like the most massive one because of all the headlines.

The flood of ongoing news coverage is understandable. It took the company a long time to figure out just how big a data leak it was dealing with, so news kept dribbling out. It turned out the company was hacked by Russian spies, giving the security breach an even grimmer outlook. The world's biggest data breach has recently returned to the front pages because of a class action lawsuit and settlement payments to affected users.

In September 2016, Yahoo! notified the public that 500 million user accounts had been breached in a 2014 cyberattack. A state-sponsored actor was suspected to be behind the attack. Just three months later, the company came forward again, saying that it discovered another breach that had occurred in August 2013. At the time, Yahoo! estimated that 1 billion user accounts had been affected, making it one of the worst data breaches of all time. After the FBI got involved, it was determined that all 3 billion Yahoo! accounts had been compromised, making it a breach of unprecedented size.

How Did It All Happen?

The hack of Yahoo! users was orchestrated by two hackers, Latvian Alexey Belan and Canadian Karim Baratov. They were hired by Russia's Federal Security Service, the FSB, to get their hands on information belonging to several high-profile persons. The hacking duo

targeted Russian journalists, employees of a Russian cybersecurity company, and Russian officials. Other targets of what turned out to be the biggest data breach ever were the CTO of a French transportation company, a Shanghai-based managing director of a US private equity firm, a Nevada gaming official, and 14 employees of a Swiss Bitcoin banking firm.

Belan and Baratov sent Yahoo! employees a series of spear-phishing emails containing a malware download link. All it took for hackers to gain access to the company's network was a single click by one staff member.

Once they were inside the network, they were to find the user database and internal tools that were used to alter the data. They quickly accomplished both goals of this major cyber attack. Making sure not to lose access to the network, they created a back door on a Yahoo! server. They returned in December 2014 to download a copy of the entire user database.

What Records Were Leaked?

Names, phone numbers, security questions and answers, as well as password recovery emails and a cryptographic value unique to each account were copied by the attackers.

What Are the Consequences of the Yahoo! Breach?

When the company realized it was a victim of the biggest hack of all time and notified everyone with a Yahoo! account, the public was shocked. A class action lawsuit ensued. Under the terms of settlement, Yahoo! has agreed to create a settlement fund worth \$117,500,000 to compensate for the damage it caused American and Israeli users. Affected individuals can opt for at least two years of credit monitoring services that include identity-theft monitoring or a cash reimbursement ranging from less than \$100 to a maximum of \$358.80. As for the perpetrators of the Yahoo! breach, Baratov was sentenced to 5 years in prison and a fine of \$2.25 million. Belan is still at large, as are the duo's FSB collaborators.

2. Collection #1-5 Data Breach

Year of breach: 2019

Data breached: 2.2 billion usernames and associated passwords

2019 kicked off with a massive data breach when the so-called Collection #1 database surfaced in mid-January. It contained a staggering mass of credentials - 773 million unique email addresses and more than 21 million passwords. The data, approximately two or three years old, was a collection of credentials acquired in previous high-profile company data breaches, including the LinkedIn and Dropbox breaches of 2016,

This was just the beginning. By the end of January 2019, four subsequent databases - Collections #2-5 - became available for free download on Torrent websites. In total, these five databases revealed 2.2 billion unique credentials. The revealed data wasn't all that sensitive - no credit card information or Social Security numbers were disclosed. But the sheer amount of data makes it one of the biggest data breaches of all time.

How Did It All Happen?

Troy Hunt, the man behind the breach notification website haveibeenpwned.com, was the first to draw public attention to the existence of Collection #1. In mid-January, several people directed him to the cloud storage website Mega, where he found the treasure trove. Collection #1 was an 87 gigabyte database containing nearly 2.7 billion rows of email addresses and passwords. Hunt went through considerable trouble to clean up the database by eliminating duplicates and stripping out unusable bits of data that were scattered in 12,000 files.

Following Hunt's notification, the database was removed from Mega - but it was still available on underground hacking forums. The information exposed in this recent data breach from early 2019 was being sold for just \$45. Before long it could be downloaded from many sites for free.

Brain Krebs, a cybersecurity journalist running his own website, got in on the investigation. He contacted the hacker selling Collection #1 and learned there were four additional leaks in the works. The remaining collections were exposed by the end of January.

What Records Were Leaked?

Collections #2-5 were like a Frankenstein's monster of information obtained in recent security breaches - Yahoo!, LinkedIn - and some not-so-recent exploits like the MySpace breach of 2008. A staggering amount of information was distributed - 845 gigabytes. In most cases, only credentials were shared. A total of 2.2 unique user names and associated passwords were exposed in all 5 collections.

What Are the Consequences of the Collection #1-5 Data Breach?

It could have been worse. Most of the information that was revealed is outdated. Years have passed since the data obtained in the MySpace breach first started circling the dark web. When it was first offered for sale, it was going for a much heftier price than \$45.

What makes this incident one of the top data breaches ever is that it opened our eyes to the fact that hackers have been sharing and saving a lot of the data breached in the earlier attacks. It is most valuable to cybercriminals who want to perform credential stuffing attacks. Given the bad habit of using the same password for multiple websites, hackers can deploy an automated process that uses email and password combinations until it gains access to a website. This puts people at risk of dangerous attacks like phishing, fraudulent loan applications, unauthorized purchases, and money transfers.

To avoid becoming a victim of future major security breaches, you should change your passwords from the standard one or two you are using to unique and complex ones for each website. Password management tools are of great assistance.

3. Aadhaar Data Breach

Year of breach: 2018

Data breached: identity and biometric information of 1.1 billion Indian citizens

The Aadhaar breach is the perfect example of a massive cybersecurity incident if ever there was one. The world's largest ID database, Aadhaar, was established by the Unique Identification Authority of India in 2009. The database contained information on more than 1.1 billion Indian citizens, including a 12-digit unique identity number, fingerprint scans of all 10 fingers, two iris scans, name, gender, and contact information.

Most Indians have an Aadhaar card even though it isn't mandatory. However, the card is required when applying for state aid or financial assistance, buying a cellular SIM card, opening a bank account, enrolling in utilities, and getting other bureaucratic things done. The news of the Aadhaar database being hacked broke in January 2018, making the biggest data breaches of 2018 list.

How Did It All Happen?

Malicious actors infiltrated the Aadhaar database through the website of a state-owned utility company named Indane. The utility provider is connected to the government database through an application programming interface that allows applications to retrieve data stored by other applications or software. Regrettably, Indane's API had no access controls. This left the company's data vulnerable. And the data of its customers, too. And every Aadhaar card owner.

Karan Saini, a New Delhi-based security researcher, discovered this system weakness and notified the state-owned company that one of the largest data breaches in history was looming. In addition to lacking access security controls, the flawed API could also allow an attacker to go through every permutation of an Aadhaar number, receiving in-depth information every time it made a hit. Saini's warnings were met with nothing but denial by the UIDAI on Twitter.

US tech portal ZDNet also got involved. Its reporters emailed Indian authorities regarding the latest security breaches in the government database several times, but to no avail. An entire month went by and they got no reply. Then the ZDNet team reached out to the Indian Consulate in New York and explained the issue to the consul of trade and customs. Two weeks passed without any action to take down the exposed database. It wasn't until March 23, 2018, after ZDNet published the story to its American audience, that Indian authorities took the vulnerable access point offline.

What Records Were Leaked?

A staggering amount of data was revealed in one of the biggest government breaches of all time. The database with information on 1.1 billion citizens had been sitting unprotected for years. In it were names, addresses, photos, phone numbers, and emails, as well as biometric data like fingerprints and iris scans. This turned out to be a credit breach too, since the database also held information about bank accounts connected with the unique 12-digit number. Before the breach was exposed, the Indian government made a tweet denying that it stored that bank information.

What Are the Consequences of the Aadhaar Breach?

The extremely poor security measures deployed by UIDAI is likely to have ongoing catastrophic consequences. Virtually all Indian adults became potential victims of identity theft and other crimes stemming from it. The worst part is - the information from this recent data breach of 2018 had already fallen into the wrong hands before the vulnerability was eliminated.

Reporters at India's Tribune newspaper were able to purchase stolen data from hackers who were offering it through a Whatsapp group. It cost only \$7 to get someone's personal information. For an additional \$4 they obtained software to print fake Aadhaar cards. The consequences of this mega breach remain to be seen.

4. First American Financial Corp. Data Breach

Year of breach: 2019

Data breached: 885 million records

First American Financial Corp., the largest title insurance provider in the US, exposed 885 million records in one of the biggest data breaches of 2019. Real-estate buyers and sellers partner up with First American to secure property transactions, sharing in-depth personal and financial information with the company. Instead of protecting the sensitive data it collects, the insurance company let it sit unprotected on its website, accessible to everyone and anyone.

How Did It All Happen?

The unguarded database was first discovered by Brian Krebs, an independent security journalist. Krebs was tipped off about this major data breach by a Washington-based real estate developer who was working with First American. He was the one who first noticed that First American's website was leaking records - potentially hundreds of thousands of them. He realized that anyone who knew the URL of a valid document on the insurance company's website could view other documents just by editing a single digit in the link. After getting no response from First American, he got in touch with Krebs. On May 24, he reported the news on his cybersecurity blog. After that, it went viral.

What Records Were Leaked?

Recent cyber attacks, like the ones targeting Marriott and Equifax, were the doing of malicious third parties who were out to get valuable data. In the case of First American, the company itself was responsible for making its records publicly available to anyone who knew where to look.

The website leaked bank account numbers and statements, mortgage and tax documents, wire transaction receipts, Social Security numbers, and drivers license images dating back to 2003. It was a gold mine for identity thieves. It is still unclear whether cybercriminals downloaded copies of the data and if so, what they plan to do with it.

What Are the Consequences of the First American Breach?

The day Kerbs published the story, the company took down the parts of its website that were spilling precious data. However, when First American made a public announcement about the events that put millions of Americans in danger of identity theft, it seriously downplayed its own responsibility in the matter. The recent data breach was characterized as “a design defect in the web application that may or may not have had an effect on the security of customer information.”

New York's Department of Financial Services immediately launched an investigation into the security failure that exposed 16 years' worth of digital documents. The U.S. Securities and Exchange Commission began looking into the matter as well, in August 2019. The results are pending.

As for the potential victims, they have filed a class action lawsuit accusing the insurance giant of failing to implement even rudimentary security measures.

5. Verifications.io Data Breach

Year of breach: 2019

Data breached: 800 million records

Among the data spills of 2019, the one affecting verifications.io took everyone by surprise - mostly because people had never heard of this company before, yet it managed to leak 800 million personal and business records. Verifications.io LLC describes itself as a “big data email verification platform.” It is hired by marketing companies to verify the validity of email addresses used in advertising campaigns. Basically, verifications.io does the heavy lifting of verifying millions of email addresses to ensure that they are active before marketers start contacting them. And then it wound up on the list of companies with data breaches.

How Did It All Happen?

In an ocean of cyber security breaches where companies are targeted by data-thirsty hackers, this incident was more of a data leak. Luckily, it was discovered by white-hat cybersecurity researchers who notified the company right away. Bob Diachenko and Vinny Troia uncovered an unprotected 150GB MongoDB database. To their surprise, it held both personal and business information. After their discovery of the cybersecurity breach, Diachenko and

Troia notified the administrators of verifications.io, who immediately took down the database. It hasn't been restored since, and the company website was deactivated too.

What Records Were Leaked?

As soon as the pair began analyzing the publicly available data that they had stumbled across, they knew they were dealing with one of the biggest data breaches of the year. The researchers found four databases containing names, email addresses, social media data, home addresses, phone numbers, gender, and birth dates. There was also delicate information about people's credit scores - characterizations like average, below average, and above average.

Other data was related to companies that could be used for generating sales leads. Company names, annual revenue figures, websites, industry identifiers for categorizing companies called SIC and NAIC, and fax numbers could be found. Diachenko and Troia got in touch with a fellow white-hatter, Troy Hunt, who runs Have I been pwned, a website where people can determine whether their data has been breached. Hunt cross-referenced the newly discovered dataset with information obtained in recent breaches and some not so recent. It turned out to be fresh.

What Are the Consequences of the Verifications.io Breach?

It is always bad when huge amounts of personal and business information become available for public download. People whose data was exposed in this cyber breach are in danger of phishing and scamming attacks. On the other hand, there were no passwords, no Social Security numbers, and no credit card information. What's more, a lot of the info contained in the verifications.io database was already publicly available. And the company took down the database right after learning about the vulnerability.

6. Equifax Data Breach

Year of breach: 2017

Data breached: 605 million records of 147 million people

The Equifax breach was colossal in many ways. First and foremost, it was gargantuan because of the extremely sensitive information that got leaked. It was huge in terms of the number of affected individuals. That's enough to make it rank high among significant security breaches.

But there's more. The notification process was slow. In the six weeks between realizing there was a breach and notifying the public, executives sold lots of Equifax stock, raising suspicions of insider trading. And it is one of the largest data breaches when it comes to the settlement fee - a staggering \$700 million.

How Did It All Happen?

Forensic analysis determined that the system was breached on March 10, 2017. The attackers exploited a vulnerability of the customer complaint portion of the Equifax website. A patch for that widely known software weakness had been released three days earlier, but Equifax's IT staff had not yet installed the update. This is how one of the biggest hacks of all time began.

Equifax's information security team didn't notice the system vulnerabilities or unpatched software despite having run a series of scans aimed at discovering them on March 15. It remains unclear why system scans failed to detect the problems. Be that as it may, the attackers stayed more or less dormant until May 2017.

Then they moved from the compromised server to other parts of Equifax's network and stole data thanks to yet another mistake. Equifax was 10 months late renewing the annual public key certificate it used to decrypt, analyze, and re-encrypt data pulled from the internal network. This lapse allowed the cyberthieves to extract terabytes of sensitive information unnoticed, landing Equifax on the list of biggest companies hacked in 2017. On July 29, system administrators finally became aware of the attack. A month of forensic investigation ensued. The public was informed of the breach on September 8.

What Records Were Leaked?

A series of slip-ups at the credit reporting agency led to the theft of approximately 605 million records belonging to 147 million Americans. Some 40% of the population got some of the following data exposed: name, date of birth, Social Security number, address, gender,

phone number, driver's licence number, email address, taxpayer ID, drivers license, and passport photo. And unlucky 200,000 individuals suffered a credit card breach too. This is information that can be used for a number of illegal activities. An identity thief could use it to open new credit card accounts or get a loan or open fake social media accounts or commit fraud posing as the victim. The possibilities are endless.

What Are the Consequences of the Equifax Breach?

The list of companies that have been hacked in the last decade is a long one. The Equifax case is unique because of the type of customer information it leaked. Affected individuals filed a class action lawsuit that was resolved in 2019. The company is to pay a total of \$700 million to damaged parties. Considering the number of people whose information was leaked, each person is entitled to a maximum settlement of \$125.

Cybersecurity specialists continue to monitor the dark web for massive dumps of the stolen data. Since none of it has appeared for sale, some believe that Chinese state actors were responsible. Their goals are thought to be espionage, not theft.

7. Facebook Data Breach

Year of breach: 2019

Data breached: 540 million records

During its 15 years of existence, Facebook has had more than its fair share of cybersecurity breaches. With approximately 2.3 billion active monthly users, Facebook collects and stores enormous amounts of data and tends to spill a lot of it, quite frequently. Maybe you remember the Cambridge Analytica scandal of 2016, for example, when the personal information of 87 million U.S. voters got exploited by consultants working on Trump's presidential campaign.

Of the two recent data breaches of 2019, uncovered within a month of CEO Mark Zuckerberg's announcement of Facebook plans to "pivot to privacy" in March 2019. The first of the security breach examples revealed that Facebook and Instagram passwords of millions of users, dating back to 2012, were left unencrypted on company servers and accessible to 20,000 Facebook employees. The second involved two third-party app developers, Cultura

Colectiva and At the Pool, and 540 million records about user tastes and preferences. We'll explore that one in more detail.

How Did It All Happen?

It's wasn't large scale cyber attacks that led to these breaches. It was the social media giant's inability to protect the massive quantities of information it collects. A team of cybersecurity investigators at UpGuard discovered two databases on Amazon's publicly accessible S3 cloud service. One belonged to a Mexican media company Cultura Colectiva, the other to a Facebook-integrated app called At the Pool. Both were available for download; both contained a social engineer's pot of gold.

Upon the discovery of the poorly configured databases on the Amazon cloud, the UpGuard team reached out to Cultura Colectiva and At the Pool. They emailed Cultura Colectiva on January 10 and again on January 14, but they never got a response. The UpGuard team then contacted Amazon Web Services in late January. AWS took note of the recent security breach and said it would inform the database owner of the incident. Two months passed and the data stayed online. It was only in April, when Bloomberg questioned Facebook about the leak, that the databases were secured. As for the database belonging to At the Pool, it was taken down during UpGuard's investigation.

What Records Were Leaked?

Data stemming from the Cultura Colectiva breach was 145GB. It consisted of more than 540 million records revealing likes, comments, reactions, account names, and Facebook IDs. The At the Pool database was smaller but contained even more detailed information: users' likes, photos, books, movies, music, friends, groups, check-in, events, interests, and 22,000 unencrypted passwords.

What Are the Consequences of This Facebook Security Breach?

The information that was available for public download on the Amazon cloud is much like the data released in the Cambridge Analytica breach. It can be used for future hacking cases by malicious actors who want to perform social engineering attacks. Or it could be used to sway an election.

The curious thing is that no matter how many breaches the social media website has and regardless of nearly constant Facebook hack news, billions of users still stay on the network. No legal action has been taken so far regarding these two user security breaches, though Facebook has been sued multiple times for violating user privacy.

8. Marriott Data Breach

Year of breach: 2018

Data breached: 500 million records

When we look at the list of recent data breaches, the one that affected the world's largest chain of hotels definitely stands out. Affecting approximately 500 million records, among them sensitive credit card information and passport numbers, it is classified as a major breach. One year after the leak was discovered, it remains unclear who was behind it. The fact that the stolen records haven't ended up on the dark web, paired with the fact that Marriott is the main hotel provider for U.S. military and government officials, focus suspicions on Chinese state-sponsored actors.

How Did It All Happen?

One of the latest data breaches, the Marriott leak was discovered in late 2018. A red flag was detected when a suspicious attempt to access the guest reservation system of Marriott's Starwood brands was made on September 8. Two days later, third-party investigators were hired to look into the incident and to help implement containment measures.

Investigators worked quickly, and on September 17, 2018, they found what caused the data leak. A remote access Trojan - a type of malware that lets hackers secretly access, monitor, and even control a computer - was used by cybercriminals in the Marriott breach. The malicious actors also deployed Mimikatz, a tool for finding combinations of usernames and

passwords in system memory. Armed with the credentials of one of the system administrators, the attackers made the suspicious guest database query on September 8. The query was caught by Accenture, the IT security company that has monitored all Starwood hotel databases since before the merger with Marriott.

Marriot purchased Starwood Hotels and Resorts in 2016. Following the acquisition, all of Starwood's corporate employees were discharged, including the staff responsible for information security. Since the Marriott booking system wasn't immediately able to handle reservations made in thousands of Starwood hotels, reservations made in those hotels continued to go through the virus-infected Starwood system. Investigators hired by Marriott discovered that the Starwood security breach happened back in 2014 and went on unnoticed for four years.

What Records Were Leaked?

During the time hackers had access to guest data, they accessed 500 million records. Full names, gender, email addresses, telephone numbers, mailing addresses, passport numbers, and credit card information were leaked. Even though the credit card numbers were protected by encryption, the encryption keys were recklessly stored on the same server that got raided by the hackers. Some of the passport numbers that got exposed in one of the biggest data breaches ever were encrypted, others were not.

What Are the Consequences of the Marriott Hack?

So far, millions of affected guests are relieved that their sensitive information, which could be used for identity theft, hasn't been posted for sale on the dark web. However, if cybercrime investigators are right to believe that the reason for the attacks was gathering intelligence on US officials, then the consequences could be much more far-reaching.

As for Marriott, most of the expenses were borne by its insurance company. By May 2019, the costs associated with the recent data breaches amounted to \$72 million. Marriott's cyber insurance policy covered \$71 million of it. The hotel industry giant was issued a \$120 million

fine by the UK Information Commissioner's office, but it has yet to pay. Several class action lawsuits have been filed by affected guests and are yet to be resolved.

2.7 Testing Technology

Black box testing approach is used to test this website by initially examining the requirements and specifications and choosing valid inputs and determining expected outputs of selected inputs.

White box testing approach is also used to test this website by understanding the functionality of the system through its source code and creating the tests cases and executing them.

3. REQUIREMENTS AND ANALYSIS

3.1 Problem Defination:

Identification of the problem is one of the obvious tasks to be performed before developing a project. A clear understanding of the problem will help build a better system and reduce the risk of project failure.

This phase consists of two main tasks:

- a) The first is to review the needs that originally initiated the project.
- b) The second is to identify at an abstract level the expected capabilities of the new system.

It helps in understanding the system properly so that all the problems are identified correctly. It also involves considering all the alternatives that exist to achieve the objectives with respect to modifying the system, even all the various ways to implement the alternatives.

After we thoroughly understood the existing system, it was concluded that all of the work was done manually. All kinds of calculations and planning were done using the human brain instead of taking advantage of the modern Information Technology

The following limitations are there in the existing system:

- Many algorithms are not available in single platform.
- Need to use different platform for different algorithm.
- Customization in decryption not available.
- Can not upload custom world list.

- Brute force, rainbow table & dictionary attack not available.
- Analyzer cost is more.

The purposed System will overcome all problems mentioned above:-

- Decrypt in few clicks.
- Multiple algorithm in single platform.
- Free vulnerability analyzer.
- Customize word list.
- Various decrypt attack type.

3.2 Requirements Specification:

3.2.1 User Requirements

1. Encryption provide data that only you can read.
2. Decryption will convert your data into plain text.
3. Analyzer will scan your web application for potential vulnerability and give you counter measures.

3.2.2 Functional Requirements

Functional desires are the services the system must supply, but the system must react to specific inputs and therefore the manner the system must behave notably state of affairs and will in addition expressly state what the system should not do.

- System will allow user to Encrypt his data in various available algorithm.

- System will allow user to brute force encrypted data to find plain text.
- System will allow creation on custom list using key words which will increase chance of decryption.
- System will allow user to upload his own custom word list for decryption.
- System will allow decryption using key.
- System will allow user to analyse his web application for sql injection, Xss & broken authentication.

3.2.3 Non-Functional Requirements

Availability: Here we are using the MYSQL as our default database so there is less chance of non-availability of it while retrieving the data.

Reliability: Website can work robustly while no loss of any information even within the part of various failures. All the personal details of the user will be stored in “Encrypted form” in order to secure the data. The system will be developed in a manner that ensures there is no malicious code or bugs.

Security: All the user details are secured because these details are stored in a password protected secure database, so there is less chance of data loss or data corruption

Maintainability: System will maintain the following data:

Plain text

Hash type

Hash value

Usability: The system is easy to handle. The user of the system can easily go through the website without any complications.

3.2.4 Problems in the Existing Application

- **Md5decrypt.net:**

Md5decrypt is fast and effective for encryption & decryption. It decrypt MD5 hash quickly compare to other platforms. But it only provide encryption & decryption for MD5 algorithm. User might need to encrypt or decrypt using other algorithms as well. In this scenario user can't use this platform.

- **Snyk.io:**

Snyk has a great and well optimized analyzer to scan web application vulnerabilities. But it's costly and doesn't scan for broken authentication & authorization.

3.3 Project Schedule:

Project scheduling provides the following benefits:

- Assists with tracking, reporting on, and communicating progress.
- Ensures everyone is on the same page as far as tasks, dependencies, and deadlines.
- Helps highlight issues and concerns, such as a lack of resources.
- Helps identify task relationships.
- Can be used to monitor progress and identify issues early.

3.4 Software and Hardware Requirements

1. HARDWARE REQUIREMENTS:

Processor: Intel(R) Core(TM) i5-5200u CPU@ 1,6 GHz.

Memory: 2GB RAM for faster speed we can use 4GB RAM.

System Type: 64bit Operating System.

2. SOFTWARE REQUIREMENTS:

FRONT END: HTML5, CSS3, JavaScript, JQuery.

BACK END: PHP7.

DATABASE: MYSQL.

3.5 Conceptual models

The student should perceive the domain of the matter and manufacture a model of the system that describes the operations which will be performed within the system and also the allowable sequences of such operations. Abstract models could comprehend complete data flowcharts, ER diagrams, object familiarized diagrams, system flowcharts, etc.

Entity relationship diagram and there description:-

- In programming building an ER demonstrate is regularly framed to speak to things that a business needs to recollect with the end goal to perform business forms. Therefore, the ER show turns into a theoretical information display that characterizes an information or data structure which can be executed in a database, normally a social database.
- Entity– relationship displaying was produced for database plan by Subside Chen and distributed in a 1976 paper. In any case, variations of the thought existed already. Some ER models demonstrate super and subtype elements associated by speculation specialization connections, and an ER model can be utilized likewise in the determination of space particular ontologies.

- **Notations:**

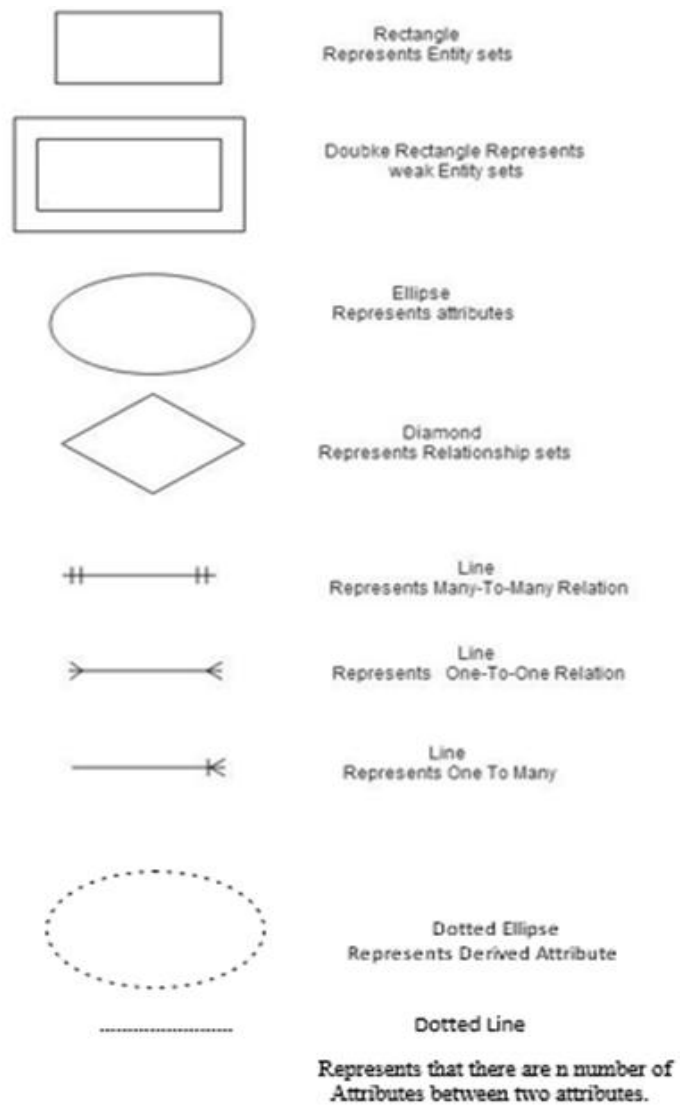
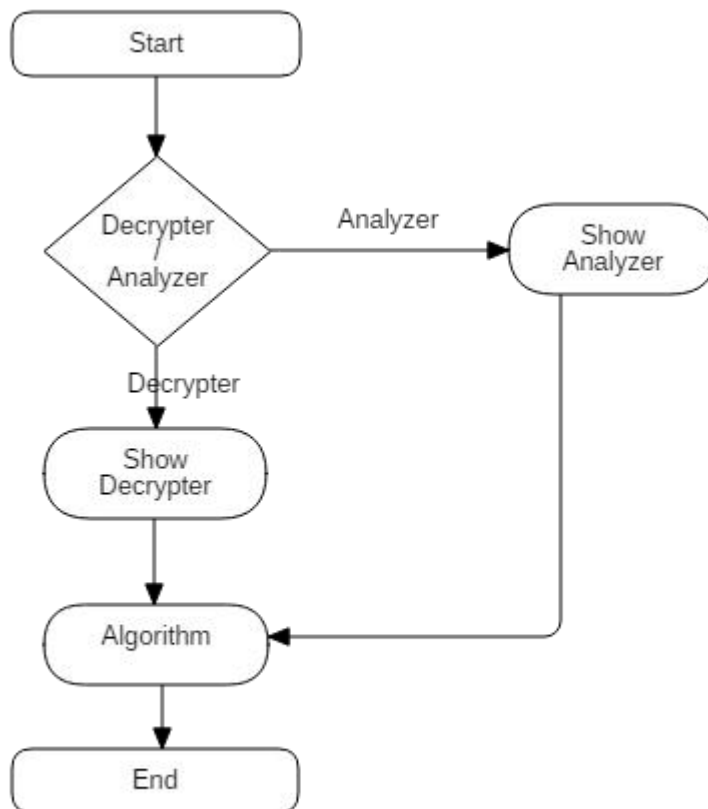
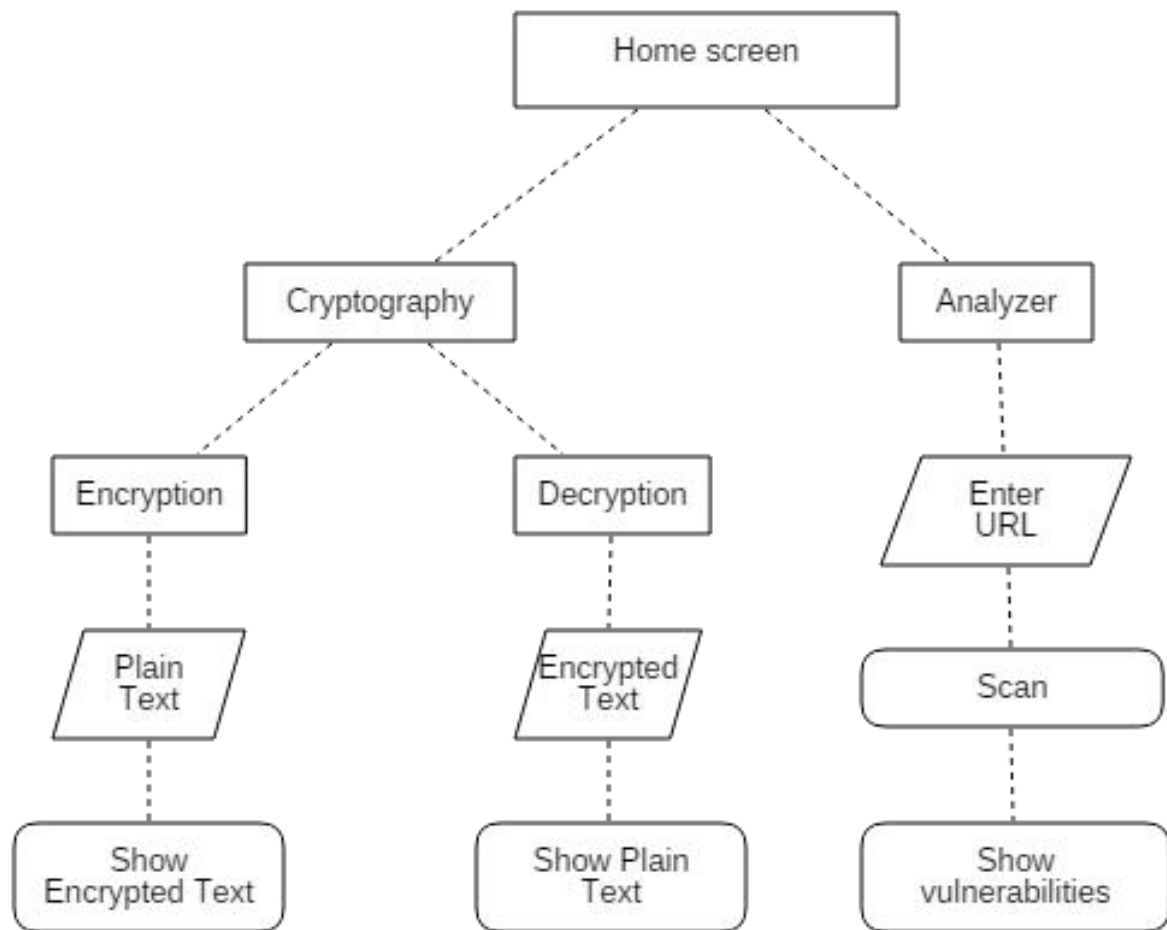


Figure 1 E-R NOTATION

- Flow of the home activity:-



On home screen user will get two options to choose. If user select Decrypter then decrypter screen will be displayed. If user select Analyzer then scanner screen will be displayed.

Normalized system diagram

4. SYSTEM DESIGN

4.1 Basic Modules:

Module 1: Encryptor

Here user will enter plain text and select encryption method and then click on encrypt button. Then based on encryption method algorithm will be selected and plain text will be converted to encrypted text.

Module 2: Decryptor

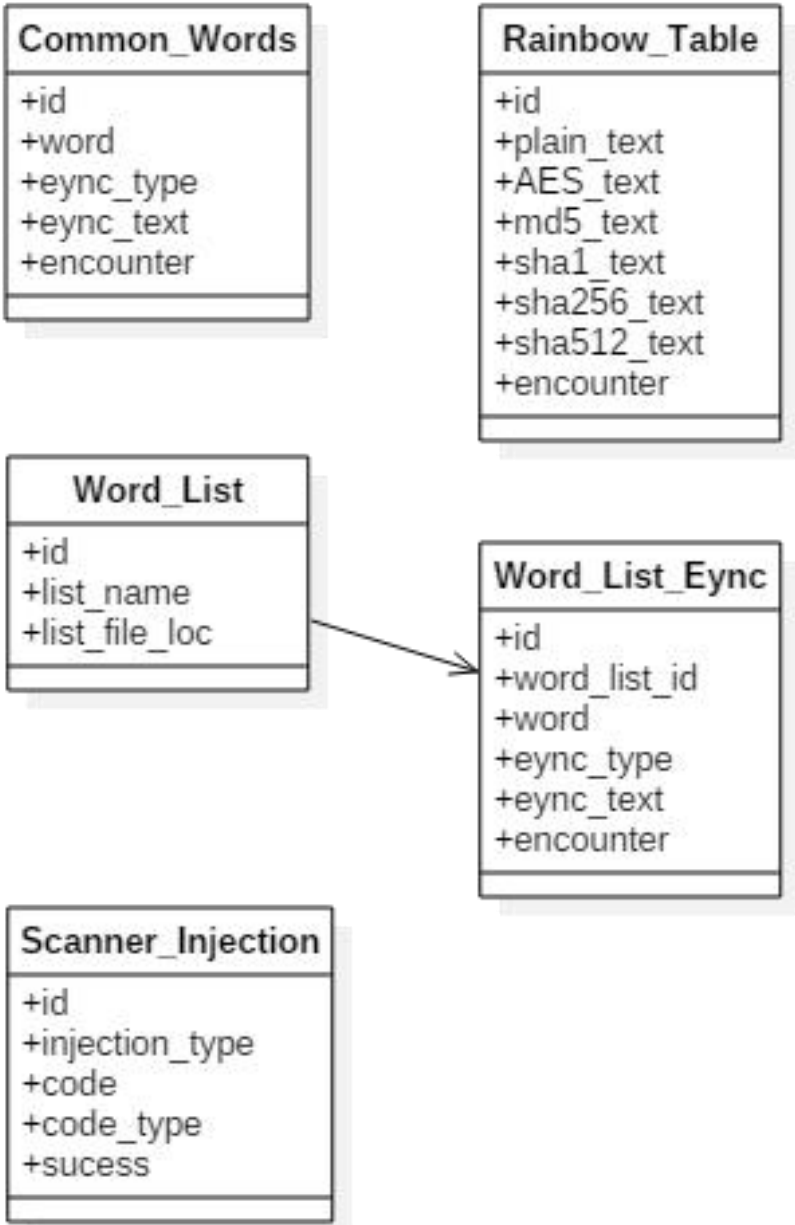
User will enter encrypted text and select decryption type and method and then click on decrypt button. Then based on method and type decryption technique will be selected and encrypted text will be converted to plain text.

Module 3: Analyzer

User will enter his web application URL and select attack type. Analyzer will scan for vulnerabilities by injecting various injections & payload in input fields and based on output vulnerabilities will be shown.

4.2 Data Design:

4.2.1 Schema Design



4.2.2 Data Integrity and Constraints:

Data integrity is the accurate and consistent entry of data throughout its lifecycle. You have no use for inaccurate or otherwise compromised data and that isn't even scratching the surface of the issues presented by the loss of sensitive data. This is why many security solutions focus so stringently on maintaining the integrity of data .

Every time data is duplicated or moved, there's the potential for alteration or loss. Each time you run a scheduled update, for instance, data has the chance to be altered. By implementing error checks and validating procedures, you can help ensure data integrity is maintained during transfers or duplications when the alteration wasn't the intention.

Rainbow table

Sr. No.	Fields	Data Fields	Allow Null	Key
1.	id	Int(10)	No	Primary Key
2.	plain_text	Varchar(200)	Yes	
3.	AES_text	Varchar(200)	Yes	
4.	Md5_text	Varchar(200)	Yes	
5.	Sha1_text	Varchar(200)	Yes	
6.	Sha256_text	Varchar(200)	Yes	
7.	Sha512_text	Varchar(200)	Yes	
8.	encounter	Int(6)	Yes	

4.3 Procedural Design:

4.3.1 Logic Diagram:

Data Flow Diagram (DFD):

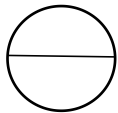
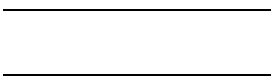
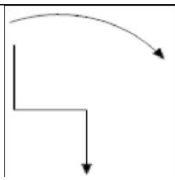

The DFD take as associate degree input method output read from the system that's knowledge objects flow within the code and square measure reworked by process component and knowledge objects from flow of the code.

Data objects displayed by tagged arrows or transformation square measure outlined by circles known as bubble and DFD is given as hierarchy fashion that's the primary knowledge flow model outlined the system as an entire ensuing DFR routine context diagram provides increasing details with every consecutive level.

Rules of DFD diagram:

1. Maintain scope of the system suggests that of context diagram.
2. Maintain DFD so main consecutive sequence of the actions.
3. Scan from left to right and prime to bottom.
4. Acknowledge all inputs or outputs.
5. Acknowledge and Label from every method internal to the system together with conic section reasonably circle.

- **Notations**

Name	Symbol	Description
Process		A method transforms incoming information flow into outgoing information flow.
Data store Notation		Data stores are same as the repositories of data in various systems.
Dataflow Notations		Data flows are pipelines through that packets of data flow. Label arrows with the context of the information that moves along with it.
External Entity Notations		External entities square measure objects outside the system,

		with that the system communicates.
--	--	------------------------------------

1. Level 0/context diagram: -

Level zero DFD may be a basic summary of the entire system or method being analyzed or modelled in Reallocation of Services Project. It's designed to be associate at-a-glance read, showing the system as one high-level method, of however the User and Admin will each move with the System.

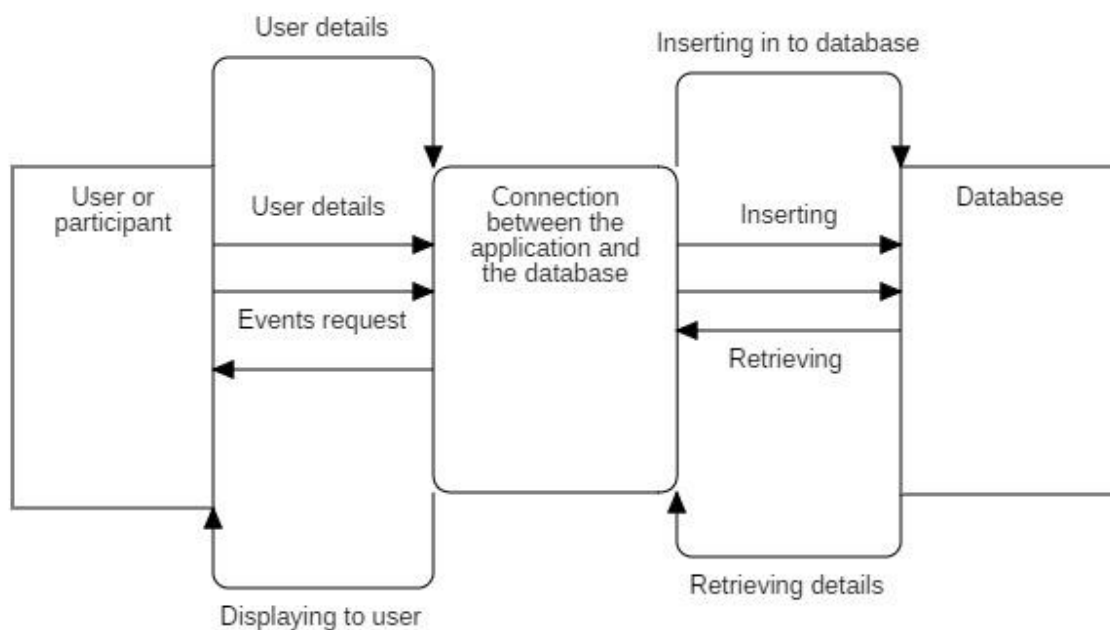


Figure 2 Context Diagram

A data flowchart (DFD) may be a graphical illustration of the "flow" of knowledge through AN data system, modelling its method aspects. A DFD is

commonly used as a primary step to make a summary of the system while not going into deep within the structure. DFDs may be used for the mental image of knowledge process (structured design).

The on top of knowledge flowchart provides the data regarding the every activity is reticulated with one another and the way the flow of the information goes once user begin the appliance. When the user begin the appliance initial time user must login within the system then it verifies whether or not the user is registered within the system or not if user verified then next activity can get open. Then our main activity opens wherever all the options of our system is displayed from there user will choose anyone. The additional elaborative description is within the abstract model section.

4.3.2 Algorithm Design:

Analyzer module algorithm:

Step 1: Open application.

Step 2: Click on analyzer.

Step 3: Enter his web application URL and select attack type.

Step 4: Scan for vulnerabilities.

Step 5: Show vulnerabilities.

Step 6: Exit.

Decryptor algorithm of the application:

Step 1: Open Decryptor.

Step 2: On dashboard user can select upload new project option.

Step 3: Enter encrypted text and select decryption type and method.

Step 4: Click on decrypt button.

Step 5: Based on method and type decryption technique will be selected.

Step 6: Show plain text.

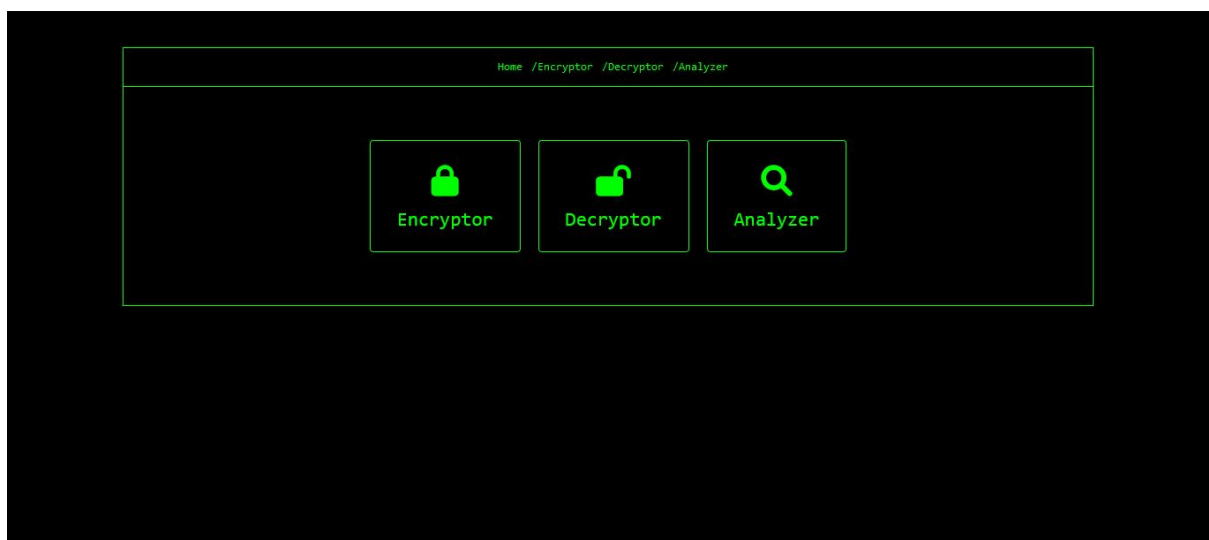
Step 7: Exit.

4.4 User Interface Design:

Design of Software is very simple which is starts with simple pages and also consists of various form which all contains different operations.

“user interface” or UI is a specialized domain within web design that's focused on bringing interaction design principles onto digital platforms. Broadly speaking, user interface design principles are often geared toward the creative end of the spectrum, with many underlying principles speaking to how a user is emotionally and functionally affected while interfacing with a specific website component.

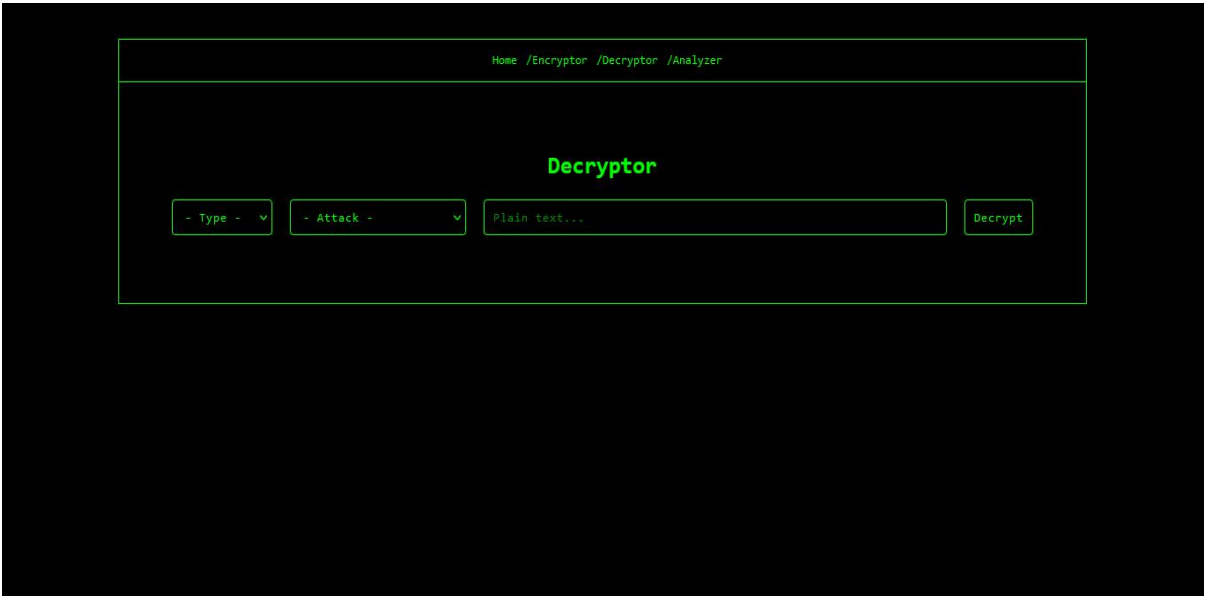
Home:



Encryptor:



Decryptor:



Analyzer:



4.5 Security Issues:

There are applications, software or more commonly used are our website consists of some sort of the authentication system which every user has to perform. So here the user will have to fill his personal details such as Name, Email-id, username, password, contact number. So this is the responsibility of that company to secure the users data at any cost because personal information is the first priority of everyone if it goes in the wrong it may lead to the dangerous consequences.

Our website provide security to the users personal credentials or more generally we can say to the personal information. This is done because of the extra care take for security.

We use different methods to keep website secure from attacks like SQL injection, XSS (Cross site scripting), CSRF (Cross Site Request Forgery), Failure to restrict URL Access, Broken Authentication, etc.

To deal with SQL we use prepared statements, So query and data can be differentiated. By using prepared statement we send data after query.

For XSS we filter every data by converting special characters to their respective HTML code during outputting data.

4.6 Test Cases Design:

Up to this section we gather the information about the system which user expect to be developed. So on that basis we jot down some test cases which we used in this and they are listed below:

Test case 1: Requirements gathering and the features to be there in the website.

Here we collect all the specification from the user which he wants to be there in the website. So if user provide all the requirements properly as per his requirement there is less chance of conflicts, because sometimes what happen that user don't tell the developer what he exactly want from the system and there it fails.

Test case 2: Technology to be used

The user who want that system is definitely not from the technical background so he did not know exactly what language and technology used in the development purpose but still sometimes the user keeps on suggesting that use that technology because it's use by the rival company. So it may mislead the development environment.

Test case 3: Scheduling of the activities and scheduling of the resources.

It basically concerns with the developing the prototype or basic module of the website and sending it to the user for the testing purpose. Based in this we priorities the activity and also allocate the resources to perform those in fix time slot.

Test case 4: User Interface

The simplicity of User Interface is person independent i.e., one user find User Interface is Complex if they haven't used any application but for another user it will be simple. Therefore, we cannot perfectly implement this requirement but we try to implement as much as possible. If the user is satisfied, then the requirement is met.

Test case 5: Response time

This application will take the less than second to perform any activity i.e. if user click on view project button, project will open immediately in other tab. Only at the uploading project section it will take some time, depending upon size of the file.

Test case 6: Cost

In this basically we check whether the all the features which user demands for can fit in the budget or not or else he will have to negotiate with the user.

CHAPTER 5:-IMPLEMENTATION AND TESTING

5.1:-Implementation Approach:-

The implementation phase is where we have to actually do the project work to produce the deliverables. The word “deliverable” means anything our project delivers. The deliverables for our project include all of the products or services that we are developing for the client, customer, or sponsor, including all the project management documents that we put together.

Once we have completed our designing and planning phase it's time for project implementation, the third phase of the project management life cycle. The implementation phase involves putting the project plan into action. It's here that we have to coordinate and direct project resources to meet the objectives of the project plan. As the project unfolds, it's our job to direct and manage each activity, every step of the way. That's what happens in the implementation phase of the project life cycle: you follow the plan you've put together and handle any problems that come up.

The means embraced to construct every deliverable will fluctuate contingent upon the sort of task we are attempted, and can't along these lines be portrayed here in any genuine detail. For example designing and media communications activities will concentrate on utilizing gear, assets, and materials to build each extends deliverable, though PC programming ventures may require the advancement and execution of programming code schedules to create each undertaking deliverable. The exercises required to manufacture every deliverable will be obviously determined inside the venture necessities record and undertaking plan.

The implementation phase keeps the project plan on track with careful monitoring and control processes to ensure the final deliverable meets the acceptance criteria set by the customer. This phase is typically where approved changes are implemented.

Most often, changes are identified by looking at performance and quality control data. Routine performance and quality control measurements should be evaluated on a regular basis throughout the implementation phase. Gathering reports on those measurements will help us to determine where the problem is and recommend changes to fix it.

5.2:-Coding Details & Code Efficiency :-

• Encryptor.php

```
<?php
$executiontime = "";
$enryctext="";
$enryctype="";
$plaintext="";
if(isset($_POST["plaintext"])){
    $enryctype = $_POST["enryctype"];
    $plaintext = $_POST["plaintext"];
    $t = microtime(true);

    $isExecuted=false;
    if($enryctype=="md5"){
        $enryctext = md5($plaintext);
        $isExecuted=true;
    }elseif($enryctype=="sha1"){
        $enryctext = sha1($plaintext);
        $isExecuted=true;
    }elseif($enryctype=="sha256"){
        $enryctext = hash('sha256',$plaintext);
        $isExecuted=true;
    }elseif($enryctype=="sha512"){
        $enryctext = hash('sha512',$plaintext);
        $isExecuted=true;
    }elseif($enryctype=="bcrypt"){
        $enryctext = password_hash($plaintext, PASSWORD_BCRYPT);
        $isExecuted=true;
    }elseif($enryctype=="snefru256"){
        $enryctext = hash('snefru256',$plaintext);
```

```

        $isExecuted=true;
    }

    if($isExecuted){
        $diff=round(((microtime(true)-$t) * 1000),4);
        $executiontime="Execition time {$diff} milliseconds";
    }
}
?>

```

• Cracktext.php

```

<?php
require_once "includes/functions.php";

ini_set('memory_limit', '256M');
ini_set('max_execution_time', 0);
$dynctype=trim($_POST['dynctype']);
$eyncertext=trim($_POST['eyncertext']);
$dyncattacktype=$_POST['dyncattacktype'];

$isFound=false;
$foundWord="";
if($dyncattacktype=="rainbowtable"){
    require_once "includes/conn.php";

    $sql = "SELECT plain_text FROM rainbow_table WHERE $dynctype = :plain_text
LIMIT 1";

    $stmt = $conn->prepare($sql);
    $stmt->bindParam(':plain_text', $eyncertext);

```

```
$stmt->execute();
$result = $stmt->fetch();
if($result){
    $isFound=true;
    $foundWord=$result["plain_text"];
}
}elseif($dyncattacktype=="disatt"){
    $fileread = file('passlist/big_pass_list.txt',FILE_IGNORE_NEW_LINES);
    foreach($fileread as $word){
        $enword=geteync($word,$dyncntype);
        if($eyncntext==$enword){
            $isFound=true;
            $foundWord=$word;
            break;
        }
    }
}elseif($dyncattacktype=="numatt"){
    $range=$_POST["dyncnumrange"];
    $j=0;
    $k=0;
    $sn=0;
    $sn=9000000;
    $i=$range-1000000;
    $k=10000000;
    if($range=="100000000"){
        $sn=10000000;
        $i=10000000;
        $k=100000000;
    }
    for($i;$i<=$k;$i++){
```

[illegible]

```

        echo "<div class='encryptioninbox'>Number Deosn't Exist Between ".$sn." -
        ".$k.", Try Different Range</div>";
    }else{
        echo "<div class='encryptioninbox'>Text Deosn't Found, Try Different
        attack</div>";
    }
}
?>

```

• Functions.php

```

<?php
function get_domain($url)
{
    $pieces = parse_url($url);
    $domain = isset($pieces['host']) ? $pieces['host'] : $pieces['path'];
    if (preg_match('/(?:P<domain>[a-z0-9][a-z0-9\-.]{1,63}\.[a-z\.]{2,6})$/i', $domain, $regs))
    {
        return $regs['domain'];
    }
    return false;
}

```

```

function geteync($plaintext,$dyncntype){
    $enycntext="";
    if($dyncntype=="md5"){
        $enycntext = md5($plaintext);
        $isExecuted=true;
    }elseif($dyncntype=="sha1"){
        $enycntext = sha1($plaintext);
    }
}

```



```

        $isExecuted=true;
    }elseif($dynctype=="sha256"){
        $enryctext = hash('sha256',$plaintext);
        $isExecuted=true;
    }elseif($dynctype=="sha512"){
        $enryctext = hash('sha512',$plaintext);
        $isExecuted=true;
    }elseif($dynctype=="bcrypt"){
        $enryctext = password_hash($plaintext, PASSWORD_BCRYPT);
        $isExecuted=true;
    }elseif($dynctype=="snefru256"){
        $enryctext = hash('snefru256',$plaintext);
        $isExecuted=true;
    }
    return $enryctext;

}

?>

```

● Portscanner.php

```

<?php
require_once "../includes/functions.php";
ini_set('max_execution_time', 0);
ini_set('memory_limit', -1);

$weburl=$_POST["weburl"];
$host = get_domain($weburl);

```

```

$ports =
array("21","22","23","25","53","80","81","110","115","135","139","143","194","443","445",
"587","1433","2525","3306","3389","5632","5900","6112");

echo "<h2>Port Scanning</h2>";

foreach ($ports as $port)
{
    $connection = @fsockopen($host, $port, $errno, $errstr, 2);

    if (is_resource($connection))
    {
        echo '<p>'. $port . ' ' . '(' . getservbyport($port, 'tcp') . ') is open.</p>' . "\n";

        fclose($connection);
    }
}
?>

```

● SubDomain:

```

<?php
error_reporting(0);

$nomer = 1;

$input = $_POST['weblink'];

$url = parse_url($input, PHP_URL_HOST);

$ch = curl_init();

curl_setopt($ch, CURLOPT_URL, "https://sonar.omnisint.io/subdomains/".$url);

curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

$output = curl_exec($ch);

curl_close($ch);

```

```
$json = json_decode($output, true);
```

```
?>
```

```
<h2>Subdomain</h2>
```

```
<table class="table table-bordered">
```

```
<tr>
```

```
<th>No.</th>
```

```
<th>List Subdomain</th>
```

```
<tr>
```

```
<?php
```

```
for($i=0; $i < count($json); $i++) {
```

```
    $target = "_blank";
```

```
    echo "<tr>";
```

```
    echo "<td>".$nomer++."</td>";
```

```
    echo "<td><a target='".$target.'" class='linkcolor'
href='http://".$json[$i]."'>".$json[$i]."</a></td>";
```

```
    echo "</tr>";
```

```
}
```

```
?>
```

```
</table>
```

```
?>
```

• Whois.php

```
<?php
```

```
require "../vendor/autoload.php";
```

```
require_once "../includes/functions.php";
```

```
$weburl = $_POST["weburl"];

use Idev\Whois\Factory;
use Idev\Whois\Exceptions\ConnectionException;
use Idev\Whois\Exceptions\ServerMismatchException;
use Idev\Whois\Exceptions\WhoisException;

echo "<h2>WhoIs</h2>";

try {
    $whois = Factory::get()->createWhois();
    $info = $whois->loadDomainInfo(get_domain($weburl));
    if (!$info) {
        print "Null if domain available";
        exit;
    }
    echo "Owner: ".$info->owner."<br>";
    echo "Registrar: ".$info->registrar."<br>";
    echo "Creation Date: ".date("d/m/Y H:i:s", $info->creationDate)."<br>";
    echo "Updated Date: ".date("d/m/Y H:i:s", $info->updatedAt)."<br>";
    echo "Expiration Date: ".date("d/m/Y H:i:s", $info->expirationDate)."<br>";
    echo "States: ".$info->states[0]."<br>";

} catch (ConnectionException $e) {
    print "Disconnect or connection timeout";
} catch (ServerMismatchException $e) {
    print "TLD server (.com for google.com) not found in current server hosts";
} catch (WhoisException $e) {
    print "Whois server responded with error '{$e->getMessage()}';"
}
```

```
?>
```

• Clickjacking.php

```
<?php
```

```
$weburl=$_POST["weburl"];
```

```
?>
```

Clickjack:
<iframe src="<?php echo \$weburl; ?>" height="100px;"></iframe>

• Cors.php

```
<?php
```

```
$url = $_POST["weburl"];
```

```
if(isset($url)) {
```

```
    $headers = getHeaders($url);
```

```
    header("Access-Control-Allow-Origin: *");
```

```
    if(count($headers) == 0) {
```

```
        die("Invalid request");
```

```
    } else {
```

```
        echo $headers[0];
```

```
    }
```

```
    foreach($headers as $header) {
```

```
        if(strpos($header, "Access-Control") !== false) {
```

```
            echo " " . $header;
```

```
        }
```

```
    }
```

```
}
```

```
function getHeaders($url, $needle = false) {
```

```

$headers = array();
$ch = curl_init($url);
curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 4);
curl_setopt($ch, CURLOPT_TIMEOUT, 4);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_VERBOSE, true);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'HEAD');
curl_setopt($ch, CURLOPT_HEADER, true);
curl_setopt($ch, CURLOPT_HEADERFUNCTION, function($curl, $header)
use(&$headers) {
    array_push($headers, $header);
    return strlen($header);
});
curl_exec($ch);
return $headers;
}
?>

```

• Headers.php

```

<?php
require_once "../includes/functions.php";

$weburl=$_POST["weburl"];
$ch = curl_init();
$headers = [];
curl_setopt($ch, CURLOPT_URL, $weburl);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

curl_setopt($ch, CURLOPT_HEADERFUNCTION,
    function($curl, $header) use (&$headers)
    {

```

```

    $len = strlen($header);
    $header = explode(':', $header, 2);
    if (count($header) < 2)
        return $len;

    $headers[strtolower(trim($header[0]))][] = trim($header[1]);

    return $len;
}
);

$data = curl_exec($ch);
echo "<h2>Server Info</h2>";

$dnssr = dns_get_record(get_domain($weburl), DNS_A + DNS_NS);

if(isset($dnssr[0]["ip"])){
    echo "IP: ".$dnssr[0]["ip"]."<br>";
}
if(isset($dnssr[0]["host"])){
    echo "Host: ".$dnssr[0]["host"]."<br>";
}

if(isset($dnssr[1]) && isset($dnssr[1]["target"])){
    echo "Nameserver 1: ".$dnssr[1]["target"]."<br>";
}
if(isset($dnssr[2]) && isset($dnssr[2]["target"])){
    echo "Nameserver 2: ".$dnssr[2]["target"]."<br>";
}
if(isset($headers["server"])){

```

```

    echo "Server: ".$headers["server"][0]."<br>";
}
if(isset($headers["x-powered-by"])){
    echo "Powered by: ".$headers["x-powered-by"][0]."<br>";
}
if(isset($headers["x-powered-by-plesk"])){
    echo "Plesk: ".$headers["x-powered-by-plesk"][0]."<br>";
}

?>

```

• Main.js

```

function copyToClipboard(elem) {
    var targetId = "_hiddenCopyText_";
    var isInput = elem.tagName === "INPUT" || elem.tagName === "TEXTAREA";
    var origSelectionStart, origSelectionEnd;
    if (isInput) {
        target = elem;
        origSelectionStart = elem.selectionStart;
        origSelectionEnd = elem.selectionEnd;
    } else {
        target = document.getElementById(targetId);
        if (!target) {
            var target = document.createElement("textarea");
            target.style.position = "absolute";
            target.style.left = "-9999px";
            target.style.top = "0";
            target.id = targetId;

```



```
        document.body.appendChild(target);
    }
    target.textContent = elem.textContent;
}
var currentFocus = document.activeElement;
target.focus();
target.setSelectionRange(0, target.value.length);

var succeed;
try {
    succeed = document.execCommand("copy");
} catch(e) {
    succeed = false;
}
if (currentFocus && typeof currentFocus.focus === "function") {
    currentFocus.focus();
}

if (isInput) {
    elem.setSelectionRange(origSelectionStart, origSelectionEnd);
} else {
    target.textContent = "";
}

$("#copied").fadeIn();

var delayMillis = 1000;

setTimeout(function() {
```

```

$("#copied").fadeOut();
}, delayMillis);
    return succeed;
}

function cheackatt(){
    var att_type=document.getElementById("dyncattacktype").value;
    if(att_type=="numatt"){
        $(".numrange").show();
    }
    else{
        $(".numrange").hide();
    }
}

function testCORS(weburl, $elem) {
    $elem.html('<div class="loading
loading03"><span>L</span><span>O</span><span>A</span><span>D</span><span>I</sp
an><span>N</span><span>G</span></div>`);

    $.ajax({
        url: weburl,
        timeout: 4000
    })
    .fail(function(jqXHR, textStatus) {
        if(jqXHR.status === 0) {
            $.ajax({
                context: weburl,
                type: "POST",
                url: "analyzerdetails/cors.php",
                data: {
                    weburl: weburl

```

```

        }
    })
    .done(function(msg) {
        if(msg.indexOf("HTTP") < 0) {
            $elem.text("CORS - doesn't exist or timed out");
        } else if(msg.indexOf("301") >= 0) {
            $elem.text("CORS - CORS header exist");
        } else {
            $elem.text("CORS - CORS doesn't exist");
        }
    });
} else {
    $elem.text("CORS - failed ");
}
})
.done(function(msg) {
    $elem.text(this.weburl + " - OK");
});
}

function analyze(){
    var weburl = $("#weburl").val();
    $(".analyzbox").fadeIn();
    getHeaders(weburl);
    getWhois(weburl);
    testCORS(weburl,$("#cors"));
    getSubdomain(weburl);
    getPorts(weburl);
    getClickJack(weburl);

}

```

- **InsertRainbowTable.php**

```
<?php
ini_set('memory_limit', '900M');
ini_set('max_execution_time', 0);
require_once "includes/conn.php";

$insert=0;
$count=0;
$fileread = file('passlist/rainbow/tuscl.txt',FILE_IGNORE_NEW_LINES);
$sql = "INSERT INTO rainbow_table (plain_text, md5, sha1, sha256, sha512, snefru256)
VALUES ";
$params="";
foreach($fileread as $word){
    $count++;
    $enword=geteync($word);
    $word=addslashes($word);
    $params= " ('".$word."', '".$enword["md5"]."', '".$enword["sha1"]."',
    '".$enword["sha256"]."', '".$enword["sha512"]."', '".$enword["snefru256"]."')";
    try{
        $finalsql=$sql.$params;
        if($conn->exec($finalsql)){
            $insert++;
        }
    }catch(PDOException $e){
        continue;
    }
}
```

```
}
```

```
function geteync($plaintext){
    $enyctext=array();
    $enyctext["md5"] = md5($plaintext);
    $enyctext["sha1"] = sha1($plaintext);
    $enyctext["sha256"] = hash('sha256',$plaintext);
    $enyctext["sha512"] = hash('sha512',$plaintext);
    $enyctext["snefru256"] = hash('snefru256',$plaintext);
    return $enyctext;
}
```

```
?>
```

5.3:-Testing approach:-

A test approach is the test strategy implementation of a project, defines how testing would be carried out. Test approach has two techniques:

Proactive - An approach in which the test design process is initiated as early as possible in order to find and fix the defects before the build is created.

Reactive - An approach in which the testing is not started until after design and coding is completed.

Different Test approaches:

There are many strategies that a project can adapt depending on the context and some of them are:

- Dynamic and heuristic approaches
- Consultative approaches
- A model-based approach that uses statistical information about failure rates.
- Approaches based on risk-based testing where the entire development takes place based on the risk

- Methodical approach, which is based on failures.
- The standard-compliant approach specified by industry-specific standards.
- Factors to be considered:
 - Risks of product or risk of failure or the environment and the company.
 - Expertise and experience of the people in the proposed tools and techniques.
 - Regulatory and legal aspects, such as external and internal regulations of the development process.
- The nature of the product and the domain.

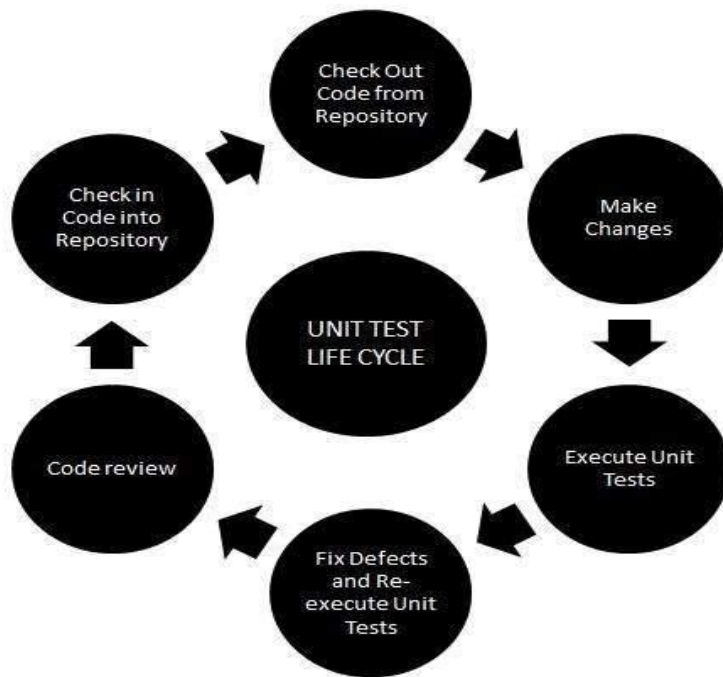
5.3.1:-Unit Testing:-

Unit testing focuses verification effort on the smallest unit of software design- the software component or module. The unit test is white-box oriented. The unit testing implemented in every module of the Online Examination System.

By giving correct manual input to the system. The data are stored in the database and retrieved. If the user needs the required module to access input or get the output from the End-user. Any error will be accrued the time will provide a handler to show what type of error will be accrued.

Unit testing, a testing technique using which individual modules are tested to determine if there are any issues by the developer himself. It is concerned with the functional correctness of the standalone modules.

The life cycle of unit testing



The main aim is to isolate each unit of the system to identify, analyze and fix the defects.

Unit Testing - Advantages:-

Reduces Defects in the newly developed features or reduces bugs when changing the existing functionality. Reduces the cost of testing as defects are captured in a very early phase.

Improve the design and allows better refactoring of code.

Unit Tests, when integrated with build gives the quality of the build as well.

Unit Testing Techniques:

- **Black Box Testing** - Using which the user interface, input, and output are tested.
- **White Box Testing** - used to test each one of those functions behavior is tested.

Gray Box Testing - Used to execute tests, risks and assessment methods

5.3.2:- Integration Testing:-

INTEGRATION TESTING is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.

Method

Any of Black Box Testing, White Box Testing and Gray Box Testing methods can be used. Normally, the method depends on your definition of 'unit'.

- **Tasks**
- **Integration Test Plan**
- **Prepare**
- **Review**
- **Rework**
- **Baseline**
- **Integration Test Cases/Scripts**
- **Prepare**
- **Review**
- **Rework**
- **Baseline**
- **Integration Test**
- **Perform**

When is Integration Testing performed?

Integration Testing is the second level of testing performed after Unit Testing and before System Testing.

Who performs Integration Testing?

Developers themselves or independent testers perform Integration Testing.

Approaches

The big bang is a way to deal with Integration Testing where all or a large portion of the units are joined together and tried at one go. This methodology is taken when the testing group gets

the whole programming in a pack. So what is the distinction between Big Bang Integration Testing and System Testing? Indeed, the previous tests just the collaborations between the units while the last tests the whole framework.

Top-Down is a way to deal with Integration Testing where top-level units are tried first and lower-level units are tried well ordered after that. This methodology is adopted when top-down improvement strategy is pursued. Test Stubs are expected to mimic lower-level units which may not be accessible amid the underlying stages.

Bottom-Up is a way to deal with Integration Testing where base dimension units are tried first and upper-level units well ordered after that. This methodology is adopted when the base up advancement strategy is pursued. Test pilots are expected to recreate larger amount units which may not be accessible amid the underlying stages. Sandwich/Hybrid is an approach to Integration Testing which is a combination of Top-Down and Bottom-Up Approaches.

5.3.3:- Beta Testing:-

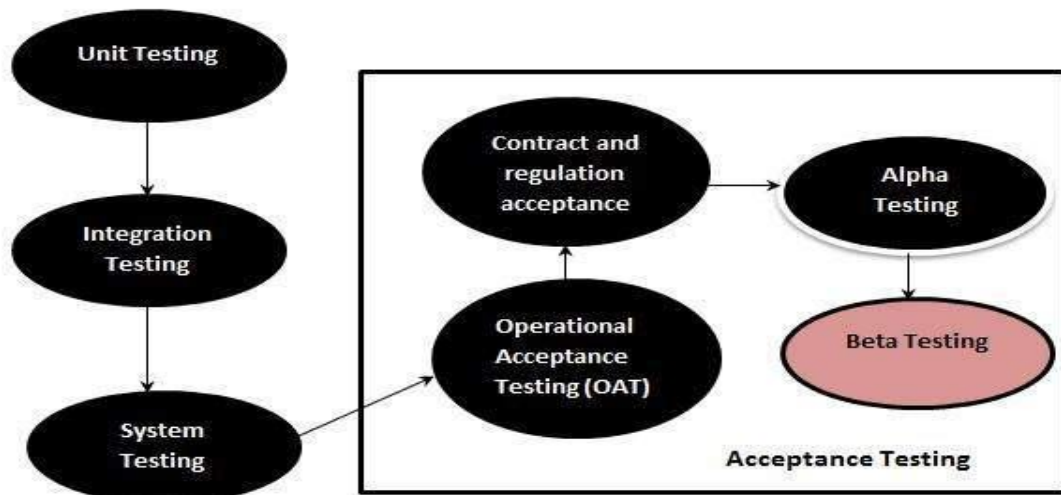
In programming advancement, a beta test is the second period of programming testing in which an examination of the target group attempts the item out.

Beta is the second letter of the Greek letters in order. Initially, the term alpha test implied the primary period of testing in a product improvement process. The principal stage incorporates unit testing, part testing, and framework testing. Beta testing can be considered "pre-discharge testing."

Beta testing is likewise now and then alluded to as client acknowledgment testing or end client testing. In this period of programming advancement, applications are exposed to true testing by the target group for the product. The encounters of the early clients are sent back to the engineers who roll out conclusive improvements before discharging the product financially.

For in-house testing, volunteers or paid guineas pigs utilize the product. For generally dispersed programming, designers may make the test adaptation accessible for downloading and free preliminary over the Web. Another reason for making programming generally

accessible along these lines is to give a see and perhaps make some buzz for the last



item.

5.4:-Modification and Improvements

After proper unit testing and integration testing, most of the major problems and errors that might occur in the future are removed.

Also, it helps us to improve the performance of the project, it helps us in memory management too i.e. while testing we search for the more efficient way of implementing the one particular unit.

CHAPTER 6:-RESULTS AND DISCUSSION

6.1:-Test Reports:-

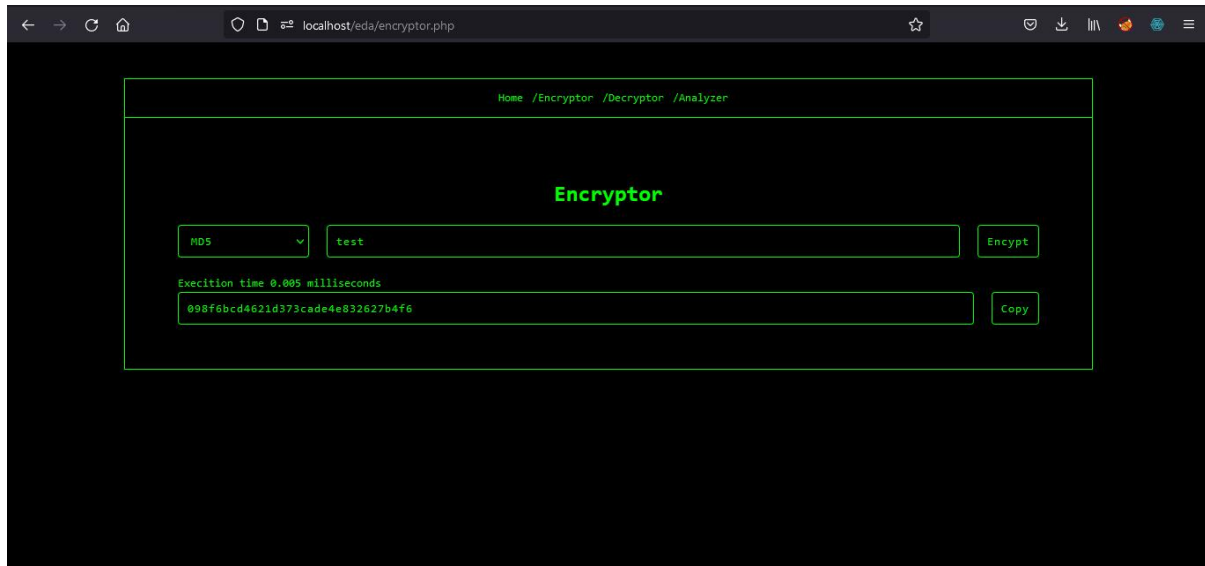
Document containing an abstract of all test activities and final test results of a testing project is popularly known as test report. Test report is a summary of how well the Testing is performed.

If the test report passes all the test preformed it is sent to the users. Based on the test report, stakeholders can evaluate the quality of the tested product and make a decision on the software release.

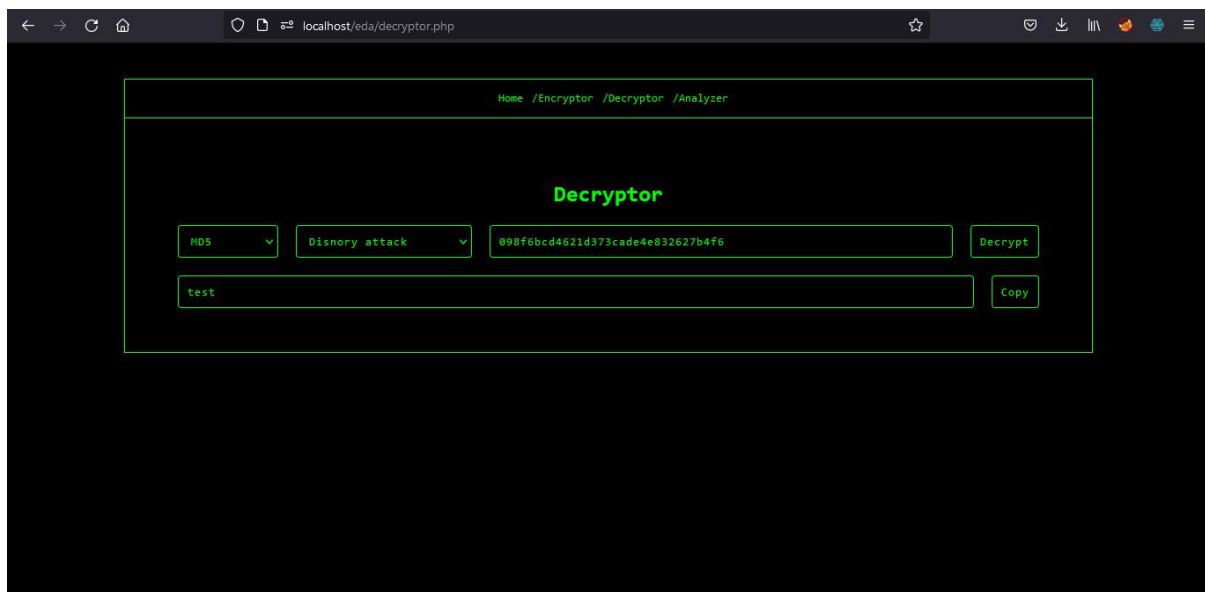
- Encryptor: When user enters the plain text it should encrypt that text.
- Decryptor: When user enter encrypted text and select attach it should decrypt it using selected attack.
- Analyzer: After entering url, it should perform various security test.
- Clickjack: Should check for Clickjack attack for given url.
- Cors: Should check if cors header is present.
- Portscanner: Should check for open ports.
- Subdomain: List all subdomains.
- Whois: Should show whois details about server.

6.2:- User Documentation

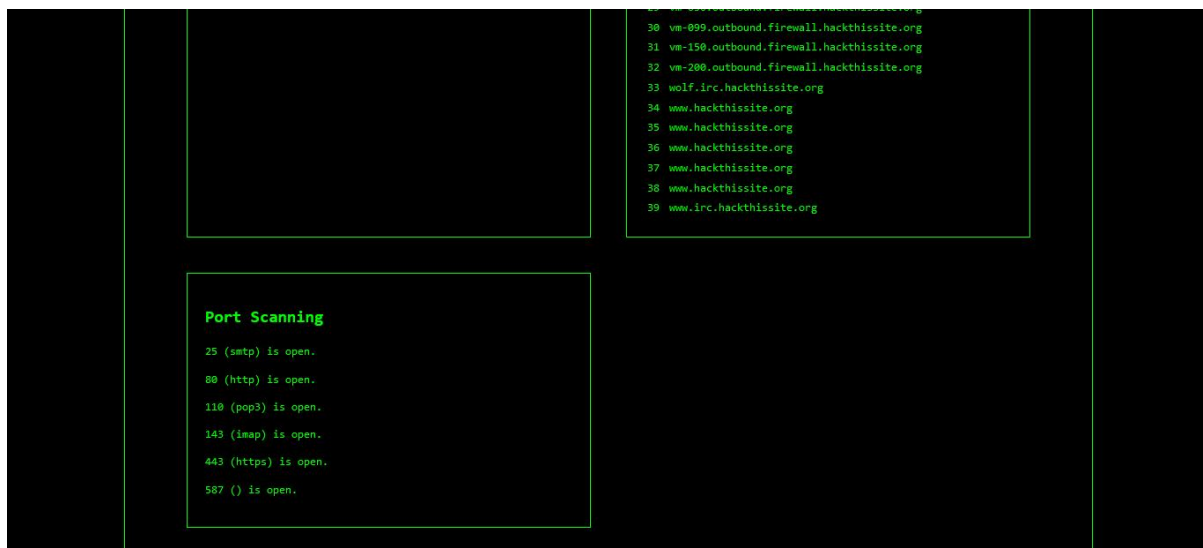
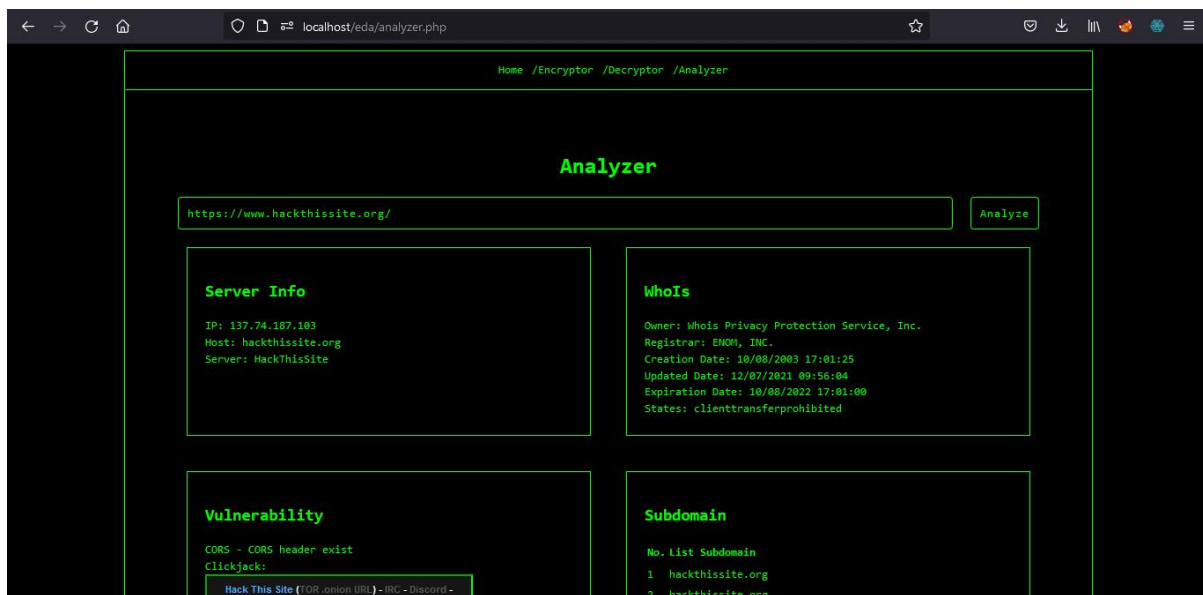
- Our encryption page where plain text is encrypted..



- Decryption page where text is decrypted using different attacks.



- Analyzer scan url for different details as shown bellow:



CHAPTER 7:-CONCLUSIONS

7.1:- Significance of the system

“Decryptor & Analyzer” is an Web application that is designed for user, developer and security tester. The main feature of this website is the attempt the decryption online. In this, the user will enter encrypted text and brute force for decryption. A vulnerability analyzer will be usefull to find loopholes in application which developer might have missed. Find vulnerabilities in your application is important before others find it.

7.2:- Limitations of the system

- Some encrypted text require too much time.
- Unable to use different scanning attacks due to limitation of web application.
- Some can be only decrypted using private key so rainbow table attack doesn't work on key & salt based encryption algorithms.

7.3:- Future scope of the project

- More encryption algorithm will be introduced.
- Key based brute force attack.
- Add more Vulnerability to scanner.

REFERENCE

- <https://stackoverflow.com/>
- <https://github.com/io-developer/php-whois>
- <https://resources.infosecinstitute.com/topic/php-build-your-own-mini-port-scanner-2/>
- <https://kitabantu.co.id/tools/Clickjacking.php>
- <https://www.youtube.com/>