B.Tech Project Preliminary Report

Secure File Accepting Prototype in Organizational Level

Submitted in partial fulfillment for the award of the Degree

of

Bachelor of Technology

in

Computer Science and EngineeringTechnology

Submitted by

Abhirami Baburaj (KTU ID : PRP17IT001)

Adrian Antony (KTU ID : KVE17CS001)

Anoj Joseph (KTU ID : PRP17CS008)

Neethu J Syam (KTU ID : LPRP17CS028)

Department of Computer Science and Engineering

COLLEGE OF ENGINEERING AND MANAGEMENT PUNNAPRA

KERALA

January 2021

B.Tech Project Report

Secure File Accepting Prototype in Organizational Level

Submitted in partial fulfillment for the award of the Degree

of

Bachelor of Technology

in

Computer Science and EngineeringTechnology

Submitted by

Abhirami Baburaj (KTU ID : PRP17IT001)

Adrian Antony (KTU ID : KVE17CS001)

Anoj Joseph (KTU ID : PRP17CS008)

Neethu J Syam (KTU ID : LPRP17CS028)

Department of Computer Science and Engineering

COLLEGE OF ENGINEERING AND MANAGEMENT PUNNAPRA

KERALA

January 2021

CERTIFICATE

This is to certify that the project report entitled **"Secure File Accepting Prototype in Organizational Level"** is a bonafide record of the project presented by Abhirami Baburaj (KTU ID : PRP17IT001), Adrian Antony (KTU ID :KVE17CS001), Anoj Joseph (KTU ID : PRP17CS008), Neethu J Syam (KTU ID : LPRP17CS028) under my supervision and guidance, in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering from the APJ Abdul Kalam Technological University for the year 2017-2021.

Sharon Jacob	Remyamol R	Smitha M Jasmine
Asst.Prof.	Asst.Prof.	Asst.Prof.
Dept. of CSE	Dept. of IT	Dept. of CSE
(Guide)	(Seminar Co-ordinator)	(HOD)

Place: Punnapra

Date: 10/01/2021

ACKNOWLEDGEMENT

I would like to place a heartfelt gratitude to the management of College of Engineering and Management,Punnapra for giving me an opportunity to excel in my efforts to complete this project on time. I am extremely grateful to **Dr.Roobin Varghese**,Principal,College of Engineering and Management,Punnapra and **Ms.Smitha M Jasmine**,HOD,Dept.of CSE,for providing all resources for the successful completion of the preliminary project presentation.My sincere gratitude to my project guide **Ms.Remyamol R**,Asst.Professor,Dept.of IT and **Ms.Sharon Jacob**,Asst.Professor,Dept.CSE for their valuable suggestions and guidance in the preparation of the project report. I express my thanks to all staff members and friends for all help and coordination extended in this project.

Abhirami Baburaj

(KTU ID :PRP17IT001)

Adrian Antony

(KTU ID :KVE17CS001)

Anoj Joseph

(KTU ID :PRP17CS008)

Neethu J Syam

(KTU ID :LPRP17CS028)

CONTENTS

LIST OF FIGURES

CHAPTER 1

OBJECTIVE

- **MODIFY THE 3 TIER ARCHITECTURE**

Information security is one of the most challenging problems in today's technological world. In order to secure the transmission of secret data over the public network (Internet), various schemes have been presented over the last decade. Most websites are build on 3 tier architecture. It Includes Web server, Application server and Data server. Usually the files are scanned only after entering application server. The modification includes the 4 tier architecture.

- **ADDITIONAL LAYER OF SECURITY**

To provide an additional layer of security between the application server and the web server we need to implement an antivirus of our own in security tier that detect various malware attacks. Basically, all servers carry AV in them, but any malware that cannot be detected by these AVs can crash the entire system and that lead to whole system failure. This can be avoided with this additional layer of security. The servers are implemented in the docker which can act as a client server and is tool designed to make it easier to create deploy and run applications by using containers.

- **STUDENT AND FACULTY DETAILS**

The dynamic web site that created should provide a user interface for students and faculty to enter their details like, name, department, register number and other regarding information in the profile. To maintain the profiles, they should make a username and a password to login. The credentials should be secure to each user and should be authenticated to login to their profile.

- **UPLOAD AND DOWNLOAD**

Uploading the assignments, notes, certificates, timetable, results should be provided in this session. The valuation of the certificates and estimating the activity points will be one main function in this session. The notes can be retrieved from the database at any time to the student and faculty.

CHAPTER 2

PROBLEM STATEMENT

- Nowadays most of the websites are build based on 3-tier architecture. That are presentation tier (web server), application tier and data tier. Usually the input files are scanned only after entering the application server. All the input files are reach the application tier even if it contains any problem. This will become a serious security issue for a website when an input file is malicious. When a malicious file is enters in the application server it would be suffer from malware. The presence of malware is yet another one of the most common threats that companies commonly have to guard against. Upon presence of malware, severe repercussions like activity monitoring, access to confidential information and backdoor access to large scale data breaches can be incurred. In most of the situations, users need to make sure to back up their important files in external safe environments. And they always need to perform checks on their security software, the browsers used, and third party plugins.

- Injection attacks come in a variety of different injection types and are primed to attack the data in web applications since web applications require data to function. The more data is required, the more opportunities for injection attack to target. SQL injection, code injection and cross-site scripting are some of the injection attacks. Injection attacks results in data leaking, removal, or manipulation of stored data. In an injection attack, an attacker supplies untrusted input to a program. This may happen when uploading an unsecured file to the website. Therefor it is important to scan all the input files before it reaches the application server for better security.

- In most situations the students are failed to submit their assignments, home works and certificates to the faculties on time if the student or the faculty is absent or they cannot meet each other. Sometimes the deadline of document submission is reached when the institution may close accidently or the student have some issues to go to the institution. Students suffers from submitting some of the categorised documents which can be submitted only to a particular faculty. Faculty cannot evaluate a document with in time when the students have late submissions.

CHAPTER 3

INTRODUCTION

Web security is an important aspect for web applications .Today web security is a real concern related to the Internet. It is considered as the principle framework for the worldwide data society. Web applications provide a better interface for a client through a web page. The web page script gets executed on client web browser.

Web applications are a main base of attacks such as cross-site scripting, cookie-session theft, browser attack, self-propagating worms in web email and web sites. These types of attacks are called 'injection attacks' which attacks by the use of malicious code. Injection attacks have commanded the highest point of web application vulnerability lists for a significant part of the previous decade.

Information security is one of the most challenging problems in today's technological world. We want to implement practice more ways to protect information by mitigating information risks. Usually most of the websites are built on three-tier Architecture.

What is a 3-Tier Architecture?

A 3-tier architecture is a type of software architecture which is composed of three "tiers" or "layers" of logical computing. They are often used in applications as a specific type of client-server system. 3-tier architectures provide many benefits for production and development environments by modularizing the user interface, business logic, and data storage layers. Doing so gives greater flexibility to development teams by allowing them to update a specific part of an application independently of the other parts. This added flexibility can improve

overall time-to-market and decrease development cycle times by giving development teams the ability to replace or upgrade independent tiers without affecting the other parts of the system.

For example, the user interface of a web application could be redeveloped or modernized without affecting the underlying functional business and data access logic underneath. This architectural system is often ideal for embedding and integrating 3rd party software into an existing application. This integration flexibility also makes it ideal for embedding analytics software into pre-existing applications and is often used by embedded analytics vendors for this reason. 3-tier architectures are often used in cloud or on-premises based applications as well as in software-as-a-service (SaaS) applications.

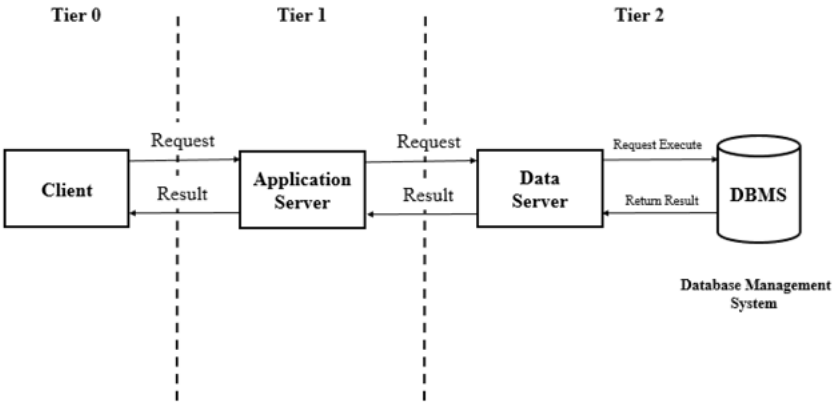


Figure 3.1: 3 Tier Architecture

What Do the 3 Tiers Mean?

Presentation Tier- The presentation tier is the front end layer in the 3-tier system and consists of the user interface. This user interface is often a graphical one accessible through a web browser or web-based application and which displays content and information useful to an end user. This tier is often built on web technologies such as HTML5, JavaScript, CSS, or through other popular web development frameworks, and communicates with others layers through API calls.

Application Tier- The application tier contains the functional business logic which drives

an application's core capabilities. It's often written in Java, .NET, C, Python, C++, etc. Data Tier- The data tier comprises of the database/data storage system and data access layer. Examples of such systems are MySQL, Oracle, PostgreSQL, Microsoft SQL Server, MongoDB, etc. Data is accessed by the application layer via API calls. The typical structure for a 3-tier architecture deployment would have the presentation tier deployed to a desktop, laptop, tablet or mobile device either via a web browser or a web-based application utilizing a web server. The underlying application tier is usually hosted on one or more application servers, but can also be hosted in the cloud, or on a dedicated workstation depending on the complexity and processing power needed by the application. And the data layer would normally comprise of one or more relational databases, big data sources, or other types of database systems hosted either on-premises or in the cloud.

A simple example of a 3-tier architecture in action would be logging into a media account such as Netflix and watching a video. You start by logging in either via the web or via a mobile application. Once you've logged in you might access a specific video through the Netflix interface which is the presentation tier used by you as an end user. Once you've selected a video that information is passed on to the application tier which will query the data tier to call the information or in this case a video back up to the presentation tier. This happens every time you access a video from most media sites.

THE EXISTING SYSTEM

A website developed for an Organization based on the three-tier architecture have the chances to get attacked. For an organization , if the system is cracked ,it is very hard to replace the whole system which is time consuming and costly.

THE PROPOSED SYSTEM

The objective of this project is to develop a system which follows a four- tier architecture. Here we add an additional layer of Security between the web sever and Application sever. This Security layer contains a Security sever where an antivirus program is in it. Therefore,

the request from the client reaches the application sever only after it was scanned by the security sever . so that the virus or any other malware enters the system will only harm the security server not the entire system . This System is developed by using Docker and antivirus installed in the security server.

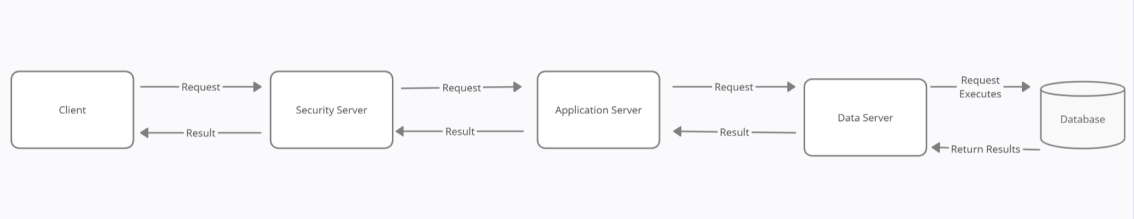


Figure 3.2: Modified Architecture

This project entitled **"Secure file accepting protocol in Organizational level"**. We modifies the system architecture as well as develop a website for showing the functioning of the system.
The website mainly have three modules

- Administrator module
- Student module
- Faculty module

The Administrator Module: The administrator module is the core module in the website. This module handles complete transactions in the website. The functions of the administrator module includes authentication of user login and removing user accounts.
The Student Module: The student module is one of the most using module in the website. The functions of the student module includes adding, updating, deleting profiles. uploading notes, certificates etc.
The faculty module: The faculty module is the other most using module in the website. The functions of the faculty module includes uploading notes , assignment questions,

question papers for examinations etc and evaluation of assignments, answer sheets and uploading corresponding marks.

CHAPTER 4

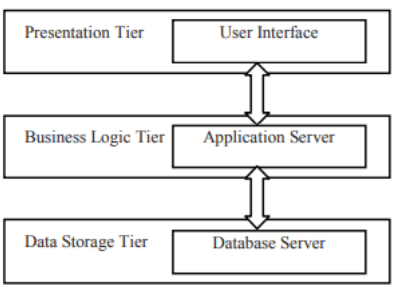
LITERATURE REVIEW

4.1 STUDY ON APPLICATION MODEL OF THREE-TIRED ARCHITECTURE (Base Paper)

Authors: Jun Tie, Jia Jin , Xiaorong Wang

This paper discusses characteristics and merits of three-tiered architecture in the system application, points out the core idea of three-tiered architecture is the distribution of calculation and data. This Paper put forward a basic principle that the middle tier should be a buffer of other tiers and the balance point of all loads, and it should be expanded for future business requirements. Based on this principle, we propose a new perspective that the middle tier in traditional three-tiered architecture should be logically divided into Basic Business Tier, Kernel Business Tier and Business Presentation Tier. Finally, the realization of this division is introduced with the example of the doctor's advice management in Hospital Management Information System.

The popular three-tiered architecture is divided into three levels in data process, as dis-



played in Figure.

–Presentation Tier

Presentation Tier is the user interface of the application, and functions as the conversation between the application and the users, providing the users' friendly access to the system. In order to facilitate the direct operation to the system, the Graphic User Interface (GUI) is frequently used.

–Business Logic Tier (Middle Tier)

It is the core of the application system to realize the business logics, which demonstrate the function of the whole system. The carrier of this tier always is the application server or web server.

–Data Storage Tier

DBMS is responsible for the data storing, access, and optimization. So it has to execute the update and the query of large amount of data promptly. At present, it is mainly about the RDBMS.

The three-tiered architecture has added an application server on the basis of the traditional two-tiered architecture, which processes the business logics separately [1]. So the user interface and the business logics are located on the different platforms. And the communication protocol between them is defined by the system, which might be CORBAEJB, and MTS/COM+. Through this kind of architecture design, all users share the business application logic. Just like placing a standardized processing factory and transition path in the middle of the two tiers, the system can produce and transmit the products clients need effectively [2]. In more details:

–The flexible hardware system configuration

Each tier could choose the suitable hardware for different processing load and features. This is a problem that is directly related to the expandability. For example, the data tier and the business logic tier are to be placed on the same server at first. But the development of business also brings about the increase of the number of the user and the amount of data. The old server could be the specific server of business logic tier by adding another server especially for the data tier. If the business and the users continue to expand, the number of server in business logic tier could be increased as well with the purpose to divide

the database. The reasonable three-tiered division could facilitate the system construction alteration.

–Easy alteration and maintenance of application technical specification

The three-tiered architecture separates the presentation part and business logic part according to the application server and client. The communication and data exchange in different tiers could be realized through the middleware or the related module.

When the business logics of the database and application server have to be changed, the clients needn't do anything, or vice versa. In a word, it can improve the reusability of the system module; shorten the development cycle; also cut the maintenance costs.

–Enhance the expansibility of system

The modularization makes the system easy to be expanded horizontally and vertically. On the one side, the system can be updated into a larger and powerful platform, on the other side; the larger dimension can strengthen the network application of system. The distributed data processing becomes possible without the restriction of system isomorphism. As a result of the encapsulation of business logic tier, the data distribution and the database is invisible to presentation tier [3].

–The strict security management

In the three-tiered architecture, the identification of user is subject to the tier, the same to the access authority of data and application. Therefore, even if the outside intruder broke through the security defense of presentation tier, the system also may prevent the intruder from entering other parts through the security protection of different tiers.

4.2 A STUDY ON WEB APPLICATION SECURITY AND DETECTING SECURITY VULNERABILITIES

Authors: Sandeep Kumar, Renuka Mahajan, Naresh Kumar, Sunil Kumar Khatri

Web security is an important aspect for web applications. Today web security is a real concern related to the Internet. It is considered as the principle framework for the worldwide data society. Web applications provide a better interface for a client through a web page .The web page script gets executed on client web browser.

Web applications are a main base of attacks such as cross-site scripting, cookie-session theft, browser attack,self-propagating worms in web email and web sites. These types of attacks are called 'injection attacks' which attacks by the use of malicious code. Injection attacks have commanded the highest point of web application vulnerability lists for a significant part of the previous decade.

There are two most common security vulnerabilities today: SQL injection and cross-site scripting. A security evaluation of application defencecenter, which had more than 250 e-commerce applications, online banking and the corporate sites came up with a statement that more than 85The main issue in web security research is in enabling a user a safe and trusted platform for communication with the web application. But some people continue to do business with insecure site. Some organizations or companies don't want to reveal the information about their own security holes. So, it's very hard task to get the reliable information about the state of web security today.

This paper explores Reasons for Attacks, methods for detecting threats and assess why they have not proven more successful.A better mechanism for minimizing such type of web vulnerabilities is proposed in this paper. Currently, there are many privacy risks in web applications. Today too many websites are hacked by anonymous people. They target website because of different types of reasons.

Stealing Sensitive Information 42% ,Defacement 23% ,Planning Malware 15% ,Unknown 08% ,Deceit 03% ,Blackmail 02% ,Link Spam 03% ,Worm 01% ,Phishing 01% .....Etc

Client: The client of a web browser is effectively making client requests for pages from servers all over the web. In thisarticle client login to system normally, client sends request to server and gets response. This happens only in normal scenerio.

Attacker : Attacker is a unauthorized user.Typicallythis kind of attacker would be a proficient programmer or engineer with sufficient technical knowledge to understand the weak points in a security system.

- SQL Injection Attack: The common use of SQL injection attack is to abuse web pages that allows users to input data into form fields for database queries. Injection is an unintended command sent to an interpreter. Attackers can enter the modified SQL query for user information. The queries directly communicate with database for operations on data like data delete, create and change .The queries create link of the static part and value intended for attack .
- Cross Site Scripting Attack(XSS) Cross site scripting(XSS) is also serious problem of web application that can be used by an attacker .The attacker can insert the malicious script in web application through any external resource. The web browser executes the malicious code as a legitimate code.

PROTECTION AGAINST SQL INJECTION ATTACKS

Malicious attacks make web applications less secure because the intruder can harm the integrity of the database by applying malicious queries. we can improve web applications performance .The developer should use this type of variable in all SQL statements and also to Java language which provides better method called prepared statement .Prepared statement also uses bind variables.To defend against the SQL injection attacks, we should avoid passing the input directly into SQL queries. Instead user should use parameterized statements or sanitized input filtered carefully

This research paper provides a complete survey of current research results under web application security. We have covered all properties of web application development, under-

stood the important security functions and properties that secure web applications should use and divided existing works into three major classes .we also discuss a few issues that still need to be considered.

4.3 SOLUTION ARCHITECTURE FOR N-TIER APPLICATIONS

Authors: Tony C Shan , Winnie W Hua

This paper defines a service-oriented Solution Architecture for N-Tier Applications (SANTA), primarily for web-based distributed systems. Most conventional Internet applications have been built on three tiers – web, application, and database tiers as described in the predominant 3-tier architectural style. However, a number of leading-edge technologies have matured, which need to be incorporated into the logical solution architecture, such as portal, process choreography, business rule engine, enterprise service bus, web services, service composition, etc. A new service- oriented model is proposed in this paper, to extend the traditional 3-tier architectural style and position the emerging technologies/products in the right places in the architecture structure. This new architecture model comprises a stack of six interrelated layers, coupled with six vertical pillars. The six layers are Access Integration, Business Process, Composite Services, Services Components, Integration Communications, and Enterprise Resources layer. The Runtime Infrastructure pillars are composed of the Operational Management, Security, and Hosting Environment pillar, whereas the Development Process pillars consist of the Application Service Frameworks, Crosscutting Aspects Patterns, and Modeling Development Tools pillar. This holistic application architecture framework is a systematic taxonomy of major technical constituents of a distributed application in a service-oriented paradigm. Part of this comprehensive model has been extensively utilized in one form or another to design various SOA solutions in different industry sectors. To effectively manage the architecture complexity and organize diverse technical assets in an n-tier application, a comprehensive model is a necessity to abstract concerns, define responsibilities, and present a holistic view of the architectural aspects in a highly structured way. The Solution Architecture for N-Tier Applications (SANTA) model introduced in this paper is a multi-layered, multi-pillared framework to facilitate architecting information system applications. It provides a comprehensive taxonomy of the architectural artifacts from both development and runtime perspectives. It builds concrete architecture solutions focused on different domains, yet keeps the agility, flexibility and adaptiveness of the overall model.

The design principles of the framework are discussed in the context. The conceptual model success in our practices. Moreover, this model is scalable and flexible for dynamic extensions and expansions, which can serve as a meta-framework to incorporate other frameworks.

4.4 MULTI-LEVEL SECURITY OF WEBSITE: A REVIEW OF USE MULTI-TIER SYSTEM WITH MANY PASSWORD ENCRYPTION METHODS.

Authors: Hind Saleem Ibrahim Harba

Multi-tier web server systems are used in many important contexts and their security is a major cause of concern. Such systems can exploit strategies. In this paper, a model was present based on three-tier architecture (Client tier, Server tier and Database tier) and applying multilevel security on it. The database server tier consists of the database and the database management system (DBMS) which has been built off-line to reduce unauthorized access to sensitive data. The Client tier is generally a web-browser that displays and processes web code. Web browsers are HTTP clients that interact with the Web servers using standard protocols); which is requests code from server and then processes the code. The Middle or application server tier consists most of the application logic. Inputs are receives from the clients tire and it interacts with the database but only the results sent to application server then to client. This achieved by using multilevel of security to protect database, using Authorization, Password Encryption.

This paper describes a method for design three-tier system and protecting its streamed data from possible security attacks. The main feature of the suggested design is its ability to provide a secure environment for real-time data or file downloading watching. One of security parts is to secure pages content, this done by used authorizing and authentication. Secondly, is to make password hashing very strong to prevent the password cracking by attacker. Thirdly, secure communication between Web browsers and servers this done by

used RC4A with Secure Sockets Layer/Transport Layer Security (SSL/TLS).

Architecture of the Proposed System

The system is constructed from three-tier (client-server architecture) ;

• Client tier: a web browser runs in any computer which is responsible for handling the information representation for user request.

• Application server tier: it resides in middle tier, where it handle the initialization and the updated information. It is responsible for receiving client request, processing the data contained in request and applying the client response for updating the demand information.

• Database tier: the backend database reside on web server side and stores the data for system, which is required by the middle tier.

System security

The security of system can be described as :

1- **Data Confidentiality**: The Client-to-Application Server Protocol (CAP) gives a standard method for multi-protocol datagrams transporting over client to application server links. The CAP uses the RC4A algorithm to provide data confidentiality. The process of sending encrypted information is from application server to client to configure secure channel and this information is decrypted in client.

2- **User Authentication and Authorization:** Site Authorization is used to determine the level of user (visitor, member or administrator). The visitor can view the books content but cannot view or buy books but only the activated members have ability to view allowable books or buying not allowable books. In order to view member library user needs to enter authentication process to redirect to the member library.

3- **Password Encryption:** This method is used to encrypt user password transfer from client to server and then storied in MySQL database to protect them from being stolen or attacked from different attacks (SQL injection, Dictionary and Brute-Force Attack, Lookup Tables, Reverse Lookup Tables Rainbow Table, etc.), this is done by hashing password using BCrypt and salt.

4- **Disabling Browser Caching:** The Method used to bypass, clear, and disable browser cache so that each time client visit a page all the files are freshly downloaded.

5- **HTTP over Secure Sockets Layer (HTTPS):** It gives authentication of the website

and associated webserver that client communicating with it. HTTPS provide a protection against Man-in-the-middle attacks. In addition, it also provides bidirectional encryption of communications between server and client that protects against tampering and eavesdropping with and/or forging the contents of the communication. Thus, it provides a sensible guarantee that when client is communicating with the website, the HTTPS ensuring that the communications contents between the visitor and site cannot be read or forged by any third party.

4.5 MEMORY-EFFICIENT SIGNATURE MARKING FOR CLAMAV ON FPGA

Authors: Tran Ngoc Thinh and Tran Trung Hieu

The growth in quantity and complexity of signatures made matching task more challenge especially on general purpose processor. In this paper, proposed an efficient architecture for matching Clam Antivirus (ClamAV) signatures on FPGA. Utilizes Bloom filter technique for filtering input data and Bloomier filter technique for one round check suspect data. The matching engine support up to 256 byte length signature and can handle both basic and regular expression signatures. This paper proposed a high-speed FPGA architecture for matching Clam Antivirus (ClamAV) signatures. Exploited Bloom Filter and Bloomier Filter (BBF) techniques that result in low resource utilization and we also proposed approaches for handling wildcard containing virus signatures. The approaches are mainly based on memory, so that the system update could be simple and fast.

PATTERN ANALYZER

• Pattern Fragmentation - The length of patterns considerably varies to hundreds of characters, for that reason, cannot implement BloomBloomier Filter (BBF) for every pattern length. This consumes too many resources and wastes memory because there are some groups of patterns of the same length which only have a few patterns. Therefore, breaks patterns into fragments, then put fragments having the same length into distinct groups

and apply BBFs for those groups.

• Pattern reconstruction - Follows thee linked list methods as to reconstruct a pattern from its fragments. There are three pieces of information are encoded in the first fragment record: Distance to second fragment, hash value of that second fragment and pattern id. When a first fragment is matched, our matching engine stores its information for comparison with upcoming match of second fragment. If next fragment length, its on-arriving relative distance as well as hash value are all equivalent to any correspondence previous record, can assume these two fragments belong to one pattern.

HARDWARE ARCHITECTURE

• Bloom -Bloomier-Filter Pattern Matching Engines - BBF-Engine is a pattern matching engine which bases on the combination of Bloom filter and Bloomier Filter. BBF-Engine maintains small onchip memory, low number of fault positives and can indicate which patterns are the candidate matches. The Character Matching Unit includes a series of Hash Modules to calculate hash value for each possible string of which length is between 1 and 128 characters. There are also 5 Bloom-Bloomier Modules (BBM) for the corresponding string lengths: 8, 16, 32, 64 and 128. The Arbiter Unit repeatedly fetches bloom-match records in Bloom-match FIFOs for processing.

• Pattern Reassembly Controller - Signature set of Clam Antivirus is preprocessed before deploying on hardware. A toolset is programmed for matching through signature set and extracting all metadata associated with each pattern. For basic signatures containing only a single pattern, we assign them with lowest Signature ID (SigID). For regex signatures containing two or more patterns and wildcards, separates these patterns, and define necessary metadata for hardware reassembly function. These metadata are stored on on-chip memory (Pattern Metadata), some registers for controlling pattern borders are stored inside Address Decoder Unit.

• Pattern Reassembly Unit (PRU) - Each PRU is in charge of tracking a dedicated regex signature. Bases on value of mask, the controller could decide suitable PRU for forwarding metadata to. If the nth bit of mask is set to 1, the pattern is expected for the position of nth of the signature. Each WPU have a shift register defining the waiting pattern for

its track. This register has only one bit set to 1 and is shifted every time the new pattern coming and wildcard constraint is verified. Bases on these registers, the controller could know which PRU is suitable for this pattern and when all patterns of the regex signature are found. Metadata of the pattern is then forwarded to WPU, and further verification process is performed.

This paper proposed a virus signature matching architecture on FPGA. The system based on Bloom and Bloomier filter approaches and could support both basic and regular expression virus signatures. Experimental results on ClamAV signature database show that the system could support IGbps throughput and is more efficient than previous approaches in term of on-chip memory density. The design mainly bases on memory and could be easily updated for new virus signatures. In near future, planned to extend the system to support all ClamAV signature database. The system will be integrated as a part of a hybrid system which is a combination of hardware and software. The hybrid system could exploit the high matching throughput FPGA-based engine while maintain the flexibility and friendly user interface of software application.

4.6 MEMORY BASED HARDWARE ARCHITECTURE TO DETECT CLAMAV VIRUS SIGNATURES WITH RESTRICTED REGULAR EXPRESSION FEATURES

Authors: Nga Lam Or, Xing Wang, and Derek Pao

PROPERTIES OF THE CLAMAV PATTERN SET

Researches on hardware regex detection methods are based on pcre patterns taken from Snort. A few major differences between the ClamAV format and the pcre format are listed below.

1. Virus signatures are binary machine codes. Each byte is shown as two hex-digits. Signatures in Snort are mostly text-based, hence, each symbol is a character. Binary data value is preceded by ‘’ in pcre.
2. In ClamAV, a double ‘??’ is used to represent a wildcard byte, and a single ‘?’ represents a low or high nibble (with 4 don’t care bits). In pcre the metacharacter ‘.’ represents the wildcard character. The meta-character ‘?’ represents match 0 or 1 time, e.g. a? means that the symbol a may be matched 0 or 1 time.
3. In pcre the Kleene closure a* means matching the character a zero or more times. The symbol * in ClamAV is equivalent to the .* (dot-star) in pcre, i.e. matching a wildcard byte 0 or more times.
4. In the ClamAV format aan-mbb means n to m wildcard bytes between aa and bb. In pcre cn, m represents repetition of the symbol c by n to m times. The repetition counts found in the Snort ruleset are typically no more than 1024, but the largest displacement count found in ClamAV is over a million. 5. Some features of pcre, such as character subclass, case sensitivity, and back reference, are not included in the ClamAV format.

PROPOSED DETECTION METHOD

By using different approaches to handle the 5 types of displacement counts. The arbitrary displacement * is equivalent to the at-least count 0-. By using the * and atleast count n- as delimiter to divide a pattern into segments. Segments that are not pure string may be subdivided into 2 or more tokens. String tokens can be detected using our P-AC and QSV string detection methods. An extended P-AC detection method called PACX is used to detect fixed-length tokens with 4 to 15 bytes that contain a string component at the front plus a small number of wildcard bytes, nibbles and/or alternate bytes. MXNFA is a more general regex detection method, and it is used to detect more complex tokens that may contain counting block(s) and other regex features. Displacement counts between tokens will be verified by the aggregation unit (AU).

• Division of segment into tokens - If a segment is a string, then it is not subdivided. Short segments (e.g. 15 bytes or less) may be detected directly by the PACX pipeline or MX-NFA unit directly. If a long segment contains displacement counts, alternate bytes and

nibbles, then the segment may be subdivided into 2 or more non-trivial tokens. If there is more than 1 choice for the merging, the option with a smaller count value is chosen. If the displacement count is an exact count and the total length of the token is between 4 to 15 bytes, then the token may be detected by the PACX pipeline.

• PACX token detection unit - The PACX method is an extension of our P-AC string detection method. There are two major differences between PACX and P-AC. First, there is no feedback path in PACX, i.e. the PACX unit only contains a linear pipeline. As a result, the length of the token that can be detected by a PACX pipeline is bounded by the length of the pipeline. Second, wildcard byte, nibble, and alternate byte values can be supported by the PACX pipeline.

• MX-NFA token detection unit - The MX-NFA regex detection method was initially developed to support intrusion detection. In principle MXNFA can also be used to detect the full virus patterns but the cost can be very high for long patterns and patterns with large exact-count and range-count. In this study, the virus signatures are divided into tokens, and the MXNFA is used to process tokens that cannot be detected by other methods. The design of the MX-NFA detection unit is refined for virus detection. The MX-NFA method is based on NFA. The transition rules of the underlying NFA are stored in an active rule table, where each rule can be enabled or disabled dynamically by the embedded hardware circuits.

• Token refinements, ID assignment and design of the AU - Out of the 6059 regex signatures 5250 patterns contain only 1 segment and 809 patterns are composed of 2 or more segments. The total number of segments is 7925, and the number of distinct segments is 7568. A segment may be shared by multiple patterns. The AU is implemented as a NFA with conventional transition rule tables and embedded processing logic. A state transition graph is constructed for the set of subdivided segments.

IMPLEMENTATION AND EVALUATIONS

The set of string tokens is divided into 3 partitions. Partition 1 is detected using the P-AC method, and partitions 2 and 3 are detected using the QSV method. Partition 1 contains mainly the short segments with 3 to 7 bytes and other segments that cannot be handled by the QSV method. The MX-NFA detection unit is configured to have 24 rows, and 12 match

units per row. Each 36-entry match unit is equipped with 2 count-modules, one at each end. When the match units in a row are cascaded, the hardware can support the detection of tokens with up to 2 at most counts. Tokens are assigned to a row if they are compatible, i.e. a row of match units can output one match event per cycle. The MX-NFA match unit only outputs an 11-bit local tid. The reference location of the token is obtained from a global counter. Outputs of the 24 rows are first merged into a stream using a competition network, and then the 11-bit local tid is used to access a token descriptor table (TDT) to obtain the 15-bit actual tid and the associated control bits (isPattern, isSegment, isFirst, isSecond). In each level of the competition network the competing entry with smaller reference location can advance to the next level.

4.7 STRING SEARCHING ENGINE FOR VIRUS SCANNING

Authors: Derek Pao, Xing Wang, Xiaoran Wang, Cong Cao, and Yuesheng Zhu

In this paper, a string searching method for very large pattern sets that would require substantially smaller amount of embedded memory is presented. In this study, the system parameters are derived based on the ClamAV pattern set. However, the proposed method is applicable to other pattern sets. For the ClamAV pattern set with 82,888 patterns, our method only requires 1.4 Mbyte of embedded memory, i.e., about 1.4 bits per character of the pattern set. The proposed method is based on quick sampling of the input stream against fixed-length pattern prefixes, and on-demand verification of variable-length pattern suffixes. The effectiveness of a string searching engine depends on the hit rate (including false positives). When a genuine virus is found, the data file concerned is tagged. How to handle the infected file will be the responsibility of the software system. If the hit rate is sufficiently low, e.g., 10^7 or lower, the workload of the general-purpose processor for verification is negligible. In our evaluation using different data files as input stream with total size that exceeds 900 Mbyte, only three matches are reported and none of them are false

positives.

PROPOSED METHOD

There are over 82K static strings in the ClamAV pattern set. The minimum and maximum pattern lengths are 4 and 392, respectively. Running its algorithm on a sequential machine is very inefficient. A prerequisite is that all patterns in the pattern set are required to have distinct fixed-length (e.g., 16-byte) prefix. This property can be ensured by the preprocessing procedure. The system will do a quick sampling of fixed length segments in the input stream. If the sampled input data segment does not match any pattern prefixes in the pattern set, then no further processing at this byte location is necessary. If the sampled segment matches the prefix of a pattern, then the system will continue to compute the checksum of the input stream data for the corresponding pattern length. If the checksum is equal to the expected value, then a match result is generated. We shall show that an overall system throughput of 1 byte per cycle can be achieved using a combination of techniques including pipelining, parallel processing, hashing, and pattern set preprocessing.

• Statistical Properties of the ClamAV Database - The ClamAV virus database main.cvd version 51 released on 14 May 2009 is used in this study. Only static strings are considered. MD5 checksums and regular expressions in the database are excluded. A total of 82,888 static strings are found in the main.db and main.ndb files. The minimum, maximum, and average pattern lengths are 4, 392, and 102 bytes, respectively. One thousand two hundred fifty eight (1.5 percent) patterns have less than 16 bytes and 81,630 (98.5 percent) patterns have 16 bytes or more. Only 172 patterns have more than 180 bytes.

• QSV Architecture - The QSV module is composed of the prefix sampling unit (PS) and the CRC checksum verification unit (CRC). Patterns processed by the QSV method are required to possess unique 16-byte prefixes. Patterns that share common 16-byte (or longer) prefixes will be divided into multiple segments with 16 bytes or more, except for the last segment, such that each long segment has distinct 16-byte prefix. Short segments with 4 to 15 bytes are handled by the P-AC module. Segments with 1 to 3 bytes are ignored by the search engine. They will be verified by the postsearching verification procedure. The aggregation unit is used to combine the partial match results (detection of individual segments) to produce the final results.

• Construction of Lookup Tables - The 128-bit input stream buffer is divided into 15 regions. The bit-selection algorithm selects 2 bits from each region. Regions R8 to R14 overlap with regions R0 to R7. This arrangement ensures that the selected bits will not be localized to a few bytes of the 16-byte prefix, and provides flexibility to the bit-selection algorithm. The hash index h1 is obtained by taking one selected bit from each region, and h2 is composed of the remaining selected bits. Dividing the 128-bit key into 15 regions also helps to simplify the bitselection circuit. For each region, we require two 16-to-1 (or 18-to-1) multiplexors and two 4-bit (or 5-bit) register that stores the offset of the selected bits within the corresponding region. Generation of h1 and h2 requires another 15 bits of storage and 15 copies of 2-bit crossbar switch. The hash function can be reprogrammed by assigning new values to the registers.

PREPROCESSING OF PATTERN SET

The success of the QSV approach is subject to two prerequisites. First, patterns in QSV are required to have distinct 16-byte prefixes. This requirement can be fulfilled by a segmentation procedure. Second, the number of CRC unit is limited. The preprocessing routine identifies exception cases that can lead to overloading of the CRC pool, and transfers the exception patterns to the P-AC module.

• Segmentation of Patterns Sharing Common Prefixes - Patterns sharing common 16-byte prefixes are divided into multiple segments with length greater than or equal to 16 bytes, except for the last segment. By arranging the patterns in ascending order, groups of patterns that share common prefixes of length greater than or equal to 16 bytes can be easily identified. The segmentation algorithm will then identify the common substrings shared by the patterns in the group and divides the patterns accordingly.

• Capacity of the CRC Pool - When the prefix sampling unit finds a prefix match and the length of the associated pattern is longer than 16 bytes, a verification command will be sent to a CRC unit. The CRC unit can be kept busy for L 14 cycles. The number of CRC units available in the system is limited.

AGGREGATION UNIT

Patterns are numbered from 1 to N, and segments are assigned IDs that starts from M,

where M>N. Hence, if the pattern ID associated with a match result is smaller than M, then the match result is sent to the output interface directly. If the pattern ID is greater than or equal to M, then the match result corresponds to a partial match (matching of a segment) of a long pattern. The AU is responsible for aggregating the partial matches to produce the final result. A transition edge in the NFA contains seven fields (current state, segment ID, segment length, next state, bitmask, TTL, and state-type). The time-to-live (TTL) counter is equal to the maximum length among all segments that appear in the outgoing edges of the corresponding next state. The state-type field indicates whether the next state is an output state, a terminal state, both or none.

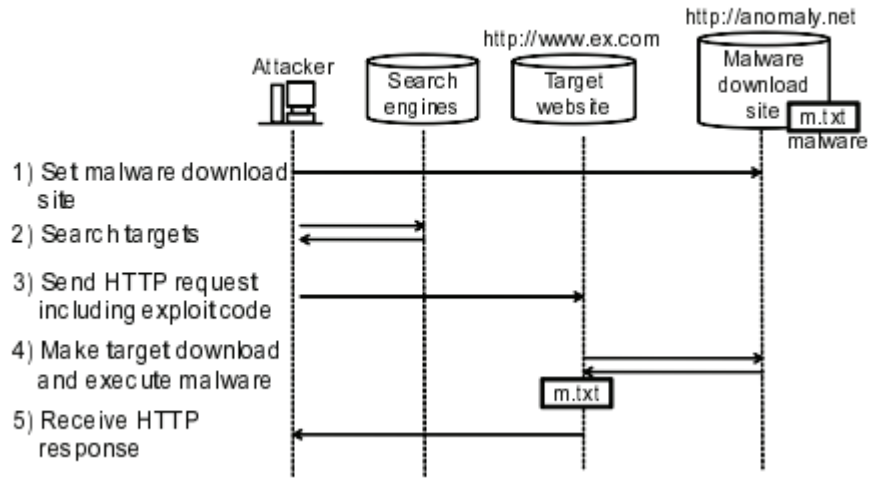
PERFORMANCE STUDY AND COMPARISON

The ClamAV virus database (main.cvd version 51 released on 14 May 2009) is used in our study. A total of 82,888 static strings are found in the main.db and main.ndb files. The minimum, maximum, and average pattern lengths are 4, 392, and 102 bytes, respectively. One hundred and fourteen duplicated strings are found. One thousand two hundred fifty eight patterns are shorter than 16 bytes, and 172 patterns are longer than 180 bytes. About 94 percent of the long patterns have distinct 16-byte prefixes. One hundred and two long patterns are found to contain prefix strings made up of repeating character or repeating short substring. About 5K (6 percent) patterns are segmented. The preprocessing routine is implemented using the C language. The program takes the raw pattern file as input and produces all the required lookup tables automatically. The execution time of the program on a PC with Intel Core2 E6400 2.13 GHz CPU is about 3 minutes. The process to determine the values of h1 and h2 requires the longest computation time. It is because the number of groups is almost doubled when one more bit is selected. This process takes about 130 seconds. The total file I/O time is about 20 seconds, and the sorting, segmentation, computation of checksums, and other analysis requires about 30 seconds.

4.8 INVESTIGATION AND ANAYSIS OF MALWARE ON WEB-SITES

Authors: Takeshi Yagi,Naoto Tanimoto,Takeo Hariu and Mitsutaka Itoh

Cyber attacks, such as phishing and scam are increasing rapidly due to growth of internet as a social Infrastructure. The malware are malicious tools programmed by attackers. Distribution of malware to legitimate user terminals and websites make illegal control on them mostly through HTTP. The attacker uses legitimate sites as hoping sites and as bots for the distribution. Force to download and execution method is used.to detect and prevent infection convolutional methods such as AV software detect the files with same characteristics as known malware. Antivirus software installed on websites receives pattern files from security vendors and detect malware by comparing the characteristics of received files with information written in the pattern files. It is difficult to detect the malware because users may turn legitimate tools to malware according to their aims. The increasing variations in malware decrease the detection ratio of AV software, since it is difficult for security vendors to provide all pattern files for each and every malware. Our project vision is to prevent the entry of the malware to the application server that may destroy the server completely. Malware collected by honeypots are used for analysing the detection ratio.



4.9 MALWARE DETECTION AND CLASSIFICATION BASED ON EXTRACTION OF API SEQUENCES

Authors: Dolly Uppal, Rakhi Sinha, Vishakha Mehra and Vinesh Jain

The developments in the IT sector have innumerable advantages but attacks on websites and computer systems are also increasing relatively. One such attack is zero-day malware attack which poses a great challenge for the security testers. The malware pen testers can use bypass techniques like Compression, Code obfuscation and Encryption techniques to easily deceive present day Antivirus Scanners. This paper elucidates a novel malware identification approach based on extracting unique aspects of API sequences. Malware attackers have made windows portable executable (PE) as their most vulnerable prey for carrying out malware-based attacks. Malware analysis can be accomplished in two ways-static analysis and dynamic analysis. Feature selection method based on N grams and odds ratio selection, capture unique and distinct API sequences from the extracted API calls thereby increasing classification accuracy. Static analysis can be defined as the method for examining software without executing it. Dynamic Analysis is defined as the method of monitoring the working of an application by analyzing its runtime performance. A model is built by the classification algorithms using active machine learning techniques to categorize malicious and benign files.

Dynamic Analysis is defined as the method of monitoring the working of an application by analyzing its runtime performance. Day by day Malware are becoming more advanced and vigilant enough so as to halt their execution as soon as they detect that their execution is simulated for security based analysis using emulators or virtual environment. In this way they can easily elude from the malware detection setups. Dynamic analysis can be preferred over static analysis because this technique does not require a deep technical understanding of the software. In addition to this, dynamic analysis is also proficient in detecting malware whose signatures are not known. In present scenario, dynamic analysis is most commonly used to detect malware but it is not adequate because there are some negative aspects of dynamic analysis like some malware exhibit malevolent characteristics of self modification

and this type of behavioral aspects cannot be unearthed using this runtime analysis technique. In addition to this, malware which depend on certain trigger conditions for their execution and can alter their behavior when these conditions are achieved, are not exposed by dynamic analysis because all possible execution paths cannot be probed in a single pass.

4.10 A COMPREHENSIVE REVIEW ON MALWARE DETECTION APPROACHES

Authors: ÖMER ASLAN AND REFIK SAMET

There have been made several studies on malware detection approaches. However, the detection of malware still remains problematic. Signature-based and heuristic-based detection approaches are fast and efficient to detect known malware, but especially signature-based detection approach has failed to detect unknown malware. On the other hand, behavior-based, model checking-based, and cloud-based approaches perform well for unknown and complicated malware; and deep learning-based, mobile devices-based, and IoT-based approaches also emerge to detect some portion of known and unknown malware. However, no approach can detect all malware in the wild. This shows that to build an effective method to detect malware is a very challenging task, and there is a huge gap for new studies and methods. Even though several new methods have been proposed by using these different malware detection approaches, no method could detect all new generation and sophisticated malware. For the known malware signature- and heuristic based detection approaches perform well.

This section investigates the problem of malware and possibility of detection. It can be said that it is impossible to design an algorithm which can detect all malware. This is because the problem of detecting the malware has shown NP-complete in many studies. This is important because before starting to build an effective detection system, it is a good practice and experience for researcher to understand the scope, limitation, and possibility of malware detector. The possibility of detection malware is remaining problematic because theoretically it is a hard problem, and practically malware creators using complicated tech-

niques such as obfuscation to make detecting process very challenging.

The new generation malware uses the common obfuscation techniques such as encryption, oligomorphic, polymorphic, metamorphic, stealth, and packing methods to make detection process more difficult. This kind of malware can easily bypass protection software that is running in kernel mode such as firewalls, antivirus software, etc. and some malware instances can also present the characteristics of multiple classes at the same time. This makes practically almost impossible to detect all malware with single detection approach. The definition of common obfuscation techniques explain as follows:

• **Encryption:** In encryption, malware uses encryption to hide malicious code block in its entire code . Hence, malware becomes invisible in the host. • Oligomorphic: In oligomorphic method, a different key is used when encrypting and decrypting malware payload . Thus, it is more difficult to detect malware which uses oligomorphic method than encryption.

• **Polymorphic:** In polymorphic method, malware uses a different key to encrypt and decrypt likewise the key used in oligomorphic method. However, the encrypted payload portion contains several copies of the decoder and can be encrypted in layered . Thus, it is more difficult to detect polymorphic malware when compared to oligomorphic malware.

• **Metamorphic:** Metamorphic method does not use encryption. Instead, it uses dynamic code hiding which the opcode changes on each iteration when the malicious process is executed . It is very difficult to detect such malware because each new copy has a completely different signature. • Stealth: Stealth method also called code protection, implements a number of counter techniques to prevent it from being analyzed correctly . For instance, it can make changes on the system and keep it hidden from detection systems.

• **Packaging:** Packaging is an obfuscation technique to compress malware to prevent detection or hiding the actual code by using encryption. Due to this technique, malware can easily bypass firewall and antivirus software. Packaged malwares need to be unpacked before being analysed. The packers can be divided into 4 different groups include compressors, crypters, protectors, and bundlers.

In this section, the limitations of malware detecting systems have been summarized. Current studies demonstrate that it is almost impossible to write an algorithm to detect all

malware. This is because the computational complexity of malware is not clear, and the detection of malware problem is proved to be NP-complete. Besides, the use of new techniques (obfuscation and packing) during malware creation also makes detection process more challenging.

MALWARE ANALYSIS

In order to understand the content and behaviours of malware, it needs to be analyzed. Malware analysis is the process of determining the functionality of malware and answers to following questions How malware works, which machines and programs are affected, which data is being damaged and stolen, etc. There are mainly two techniques to analyze malware: static and dynamic. Static analysis examines the malware without running the actual code. On the other hand, dynamic analysis examines the malware behaviors while running its code. Malware analysis starts with basic static analysis and finishes with advanced dynamic analysis. The malware is analyzed by using reverse engineering [20] and some other malware analysis tools to represent the malware in different format. Reverse engineering process

MALWARE CLASSIFICATION

Machine learning (ML) is a set of algorithm that correctly estimates the outcomes of the applications without being explicitly programmed. The purpose of the ML is to convert the input data into acceptable value intervals by using statistical analysis. By using ML, many operations can be performed on related data such as classification, regression and clustering. ML algorithms have been used in malware detection for many years. Well-known ML algorithms are Bayesian network (BN), naive Bayes (NB), C4.5 decision tree variant (J48), logistic model trees (LMT), random forest tree (RF), k-nearest neighbour (KNN), multilayer perceptron (MLP), simple logistic regression (SLR), support vector machine (SVM), and sequential minimal optimization (SMO). These algorithms are used especially in behaviour-based detection and some of other detection approaches. Although each algorithm has its own advantages and disadvantages, it cannot be concluded that one algorithm is more efficient than another. However, an algorithm can perform better than other algorithms in terms of the distribution of the data, number of features, and dependencies between properties.

OBJECTIVES

• MODIFY THE 3 TIRE ARCHITECTURE

Information security is one of the most challenging problems in today's technological world. In order to secure the transmission of secret data over the public network (Internet), various schemes have been presented over the last decade. Most websites are build on 3 tier architecture. It Includes Web server, Application server Data server. Usually the files are scanned only after entering application server. The modification includes the 4 tier architecture.

• ADDITIONAL LAYER OF SECURITY

To provide an additional layer of security between the application server and the web server we need to implement an open source antivirus clam AV that detect various virus, trojans, malware etc. Basically, all servers carry AV in them, but any malware that cannot be detected by these AVs can crash the entire system and that lead to whole system failure. This can be avoided with this additional layer of security. The servers are implemented in the docker which can act as a client server and is tool designed to make it easier to create deploy and run applications by using containers.

• STUDENT AND FACULTY DETAILS

The dynamic web site that created should provide a user face for students and faculty to enter their details like, name, department, register number and other regarding information in the profile. To maintain the profiles, they should make a username and a password to login. The credentials should be secure to each user and should be authenticated to login to their profile.

• MARKING ATTENDENCE

The faculty could upload the attendance to the specific subjects and the students can view their attendance for their subjects. The website should give this session.

• UPLOAD

Uploading the assignments, notes, certificates, timetable, results should be provided in this session. The valuation of the certificates and estimating the activity points will be one main function in this session. The notes can be retrieved from the database at any time to the student and faculty.

4.11 CONTENT MANAGEMENT SYSTEM: COMPARATIVE STUDY

Authors: Manish Nath, Anuja Arora

In this paper, the various features essential for a content management system are explored. This includes a comparison of various open source CMS , their features and a conclusion on improvements that will be made in order to alleviate problems . Our aim was to find out the best available open source CMS and implement it for our website. For this we conducted a study on 13 java based open source CMS available taking into account 29 features like the technology used for creation of the CMS and features like anti plagiarism , Digital Asset Management(DAM) , etc. for user convenience .

Selection Criteria and availability status

- Automatic UI generation: User interface generated automatically for a modification in code. The best suited is hoarder from the one's surveyed.
  Availability: 4

- Dynamic control: Dynamic control and handling of data provided. This is present in mostly all products surveyed.
  Availability: 13

- Automatic user signup: Present only in one product . Allows the user to sign in automatically at the time of opening of the site.
  Availability: 1

- Forgotten password notification: This feature supports users who have forgotten there password. This is a notification sender of a new random password. This is present in only one product.
  Availability: 1

- Email confirmation: Present in only one product this feature confirms a new user id to the user.
  Availability: 1

- Backup: Data backup facility for a user to save his data. The cms manages the data and saves it through orm. The best available product for this feature is atleap
  Availability: 4

- Navigator: Site navigator provides a map to the user showing his current location on the site map. The best product for this feature is honocms
  Availability: 2

- Admin Control: Feature of high importance. Admin control is a must for a secured cms . The best product for this feature is atleap.
  Availability: 5

- Content management: A necessary feature and present in all CMS.
  Availability: 12
  Digital Asset: Manages mpeg and avi files from the user. Provided only in one product (honocms).
  Availability: 1

- User manager: Manages the user account information and authentication .The best product for this feature is honocms.
  Availability: 5

- Template: Manages previously stored templates and provides it to the user for dynamic interfacing. The best product for this feature is honocms.
  Availability: 3

- Platform Independence: Platform independent code. All products are platform independent written in java.
  Availability: 9

- Multiple Database repositories: multiple storage of database insures backup and safety from crashes. Honocms is the best product available for this feature.
  Availability: 5

- Code auto generation: auto generation of code on drag and drop facility for the interface. The management of code is present in cms. Present in only one product honocms.
  Availability: 2

- Multiple users: multiple user access management through locking facility. The best available product is atleap for this feature.
  Availability: 8

- File uploads: management of files uploaded by the user. The best product available is atleap for this feature.

Availability: 7

- File download: management of file downloads by a user. The best product is atleap for this feature.
  Availability: 6

- Document management: management of private documents of the user for his ongoing work. The best product is atleap for this feature.
  Availability: 5

- Comments: facility to provide comments. Feature present only in roller web logger.
  Availability: 2

- Web content management: management of web content like web pages, images, etc. the best product is atleap for this feature.
  Availability: 6

- Spring: framework to impart easy coding functionality provided through IOC and AOP. Atleap is the best available product for this feature. Availability: 2

- Hibernate (ORM): object relational mapping tool to take care of database burdens. at leap is the best available product for this feature.
  Availability: 2

- Search (LUCENE): search of web content, files, classes, services etc. within the system. The best product is atleap for this feature.
  Availability: 5

- JSF: java server faces is a framework to simplify development of ui . The best product is atleap for this feature. Availability: 3

- Contact us: facility to contact the site maintenance in case of any query. Present in atleap only.
  Availability: 1

- Forums: forums for discussion by the user. Present in atleap only.
  Availability: 1

- Site map: site map to guide the user through the site. Best product is atleap for this feature.
  Availability: 1

It can be clearly seen that technologies like Hibernate , Spring and features like DAM, Anti plagiarism were absent in most CMS surveyed.We propose Alfresco as the best available open source CMS. We have successfully implemented Alfresco WCM for our website.
We have defined the workflow which keeps schedule of the work in progress.
We have defined users and roles for users to add delete update or modify content within the website. We have defined different sandboxes i.e. user sandbox and staging sandbox where in the user copy and published copy is different.
Each user can modify data according to his access rights.
Admin can lock any part of the content where he doesn't want any modification. That content becomes read only for the rest users.
Admin has full control over each user and his activity.
Only Admin assigns roles to users. While modifying a particular file the file is downloaded to the user system via user sandbox and the user can check in the file again after modification to the staging area.
Any file can be uploaded to user sandbox by a user depending on his access rights as defined

by the admin. These can be submitted to the workflow

Lucene Search index is generated for the content and users to search within the system.

4.12 MODELING THREE-TIERED WEB APPLICATION

Authors: Xue Liu, Jin Heo and Lui Sha

The rapid advancement and deployment of Web applications call for a precise yet simple model for capacity planning and analysis purposes. The most widely deployed Web application architecture is the 3- tiered system, which is composed of a front-end Web server, an application server and a backend database server. In this paper, we present an analytical model of the 3-tiered Web application architecture. We show by using queueing network theory, we can model the 3- tiered Web application architecture accurately. A testbed is built to measure model parameters based on industry standard server components and TPC-W benchmark. Validation results show that the proposed model predicts performance measures such as response time and throughput accurately.

Table 2. Model predicted and measurement data collected from testbed

N	1	5	10	50	100	150	200
M^{WS}	50	50	50	50	50	50	50
L^{WS}	0.5817	4.2548	9.1394	49.0384	99.0240	148.6923	198.9567
X^{WS}	11.1391	23.5220	27.0849	28.8355	29.2947	28.7636	27.5298
D^{WS}	0.0012	0.0014	0.0016	0.0016	0.0014	0.0015	0.0015
M^{AS}	50	50	50	50	50	50	50
X^{AS}	11.1186	23.4356	26.9335	28.7792	29.2895	28.6878	27.4815
D^{AS}	0.0151	0.0971	0.2798	1.6507	1.6749	1.7048	1.7895
M^{DS} (rounded value)	0.8923 (1)	3.1394 (3)	5.2740 (5)	25.6778 (26)	26.1442 (26)	25.4230 (25)	25.7788 (25)
X^{DS}	17.8217	33.6850	38.5728	41.4092	42.1601	41.4999	39.9598
D^{DS}	0.0254	0.0471	0.0353	0.0320	0.0298	0.0309	0.0340
D^{DS}_{norm}	0.0367	0.0679	0.0506	0.0460	0.0429	0.0448	0.0495
T (measured)	0.0531	0.1664	0.3353	1.7019	3.3948	5.2089	7.3343
T (model predicted)	0.0417	0.1527	0.3191	1.8339	3.3148	5.0793	7.1230
X (model predicted)	13.0463	26.6437	28.2397	26.7543	29.8524	29.3297	27.9407

In each individual test of this test case group, the number of EBs is fixed and each run lasts 200 seconds after the 60 seconds of warm-up period. The notations and meanings of

model parameters are summarized in Table 1. In all these tests, the think time of TPC-W EBs is set to 0.035 sec. The test results with calculated model predicted performance data are shown in Table 2. Note each transaction in TPC-W contains one HTTP request and contributes to one request at the Web server, one servlet request at the application server, but may contribute one or several SQL requests at the database server. To compensate this effect, we normalize the measured average per request service time at database. From the experiments' result shown in Table 2, we can see that the analytic model does predict response time (T) and throughput of the system (X) accurately. The small difference between the model predicted values and measured values can be due to two reasons: First, the service times at each tier are not exponentially distributed as assumed by the analytical model. Second, we used the approximated MVA algorithm for the multi-station

In this paper, we presented a queueing network model of the 3-tiered Web application architecture. In this model, the server at each tier is modeled as a multi-station queueing center, which represents the multi-threaded architecture commonly structured in modern Web applications. Validation tests show the proposed model can predict performance measures such as response time and throughput accurately.

CHAPTER 5

DESIGN

5.1 Use-Case Diagram

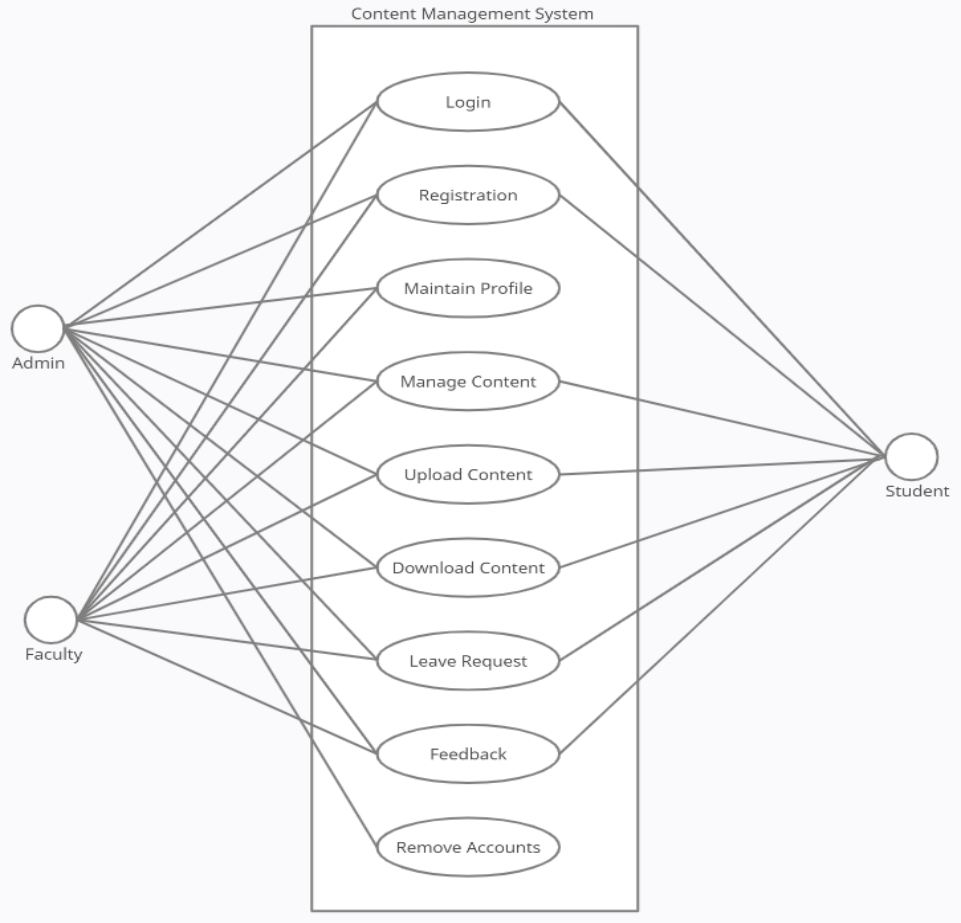


Figure 5.1: Use-Case diagram

5.2 Activity Diagrams

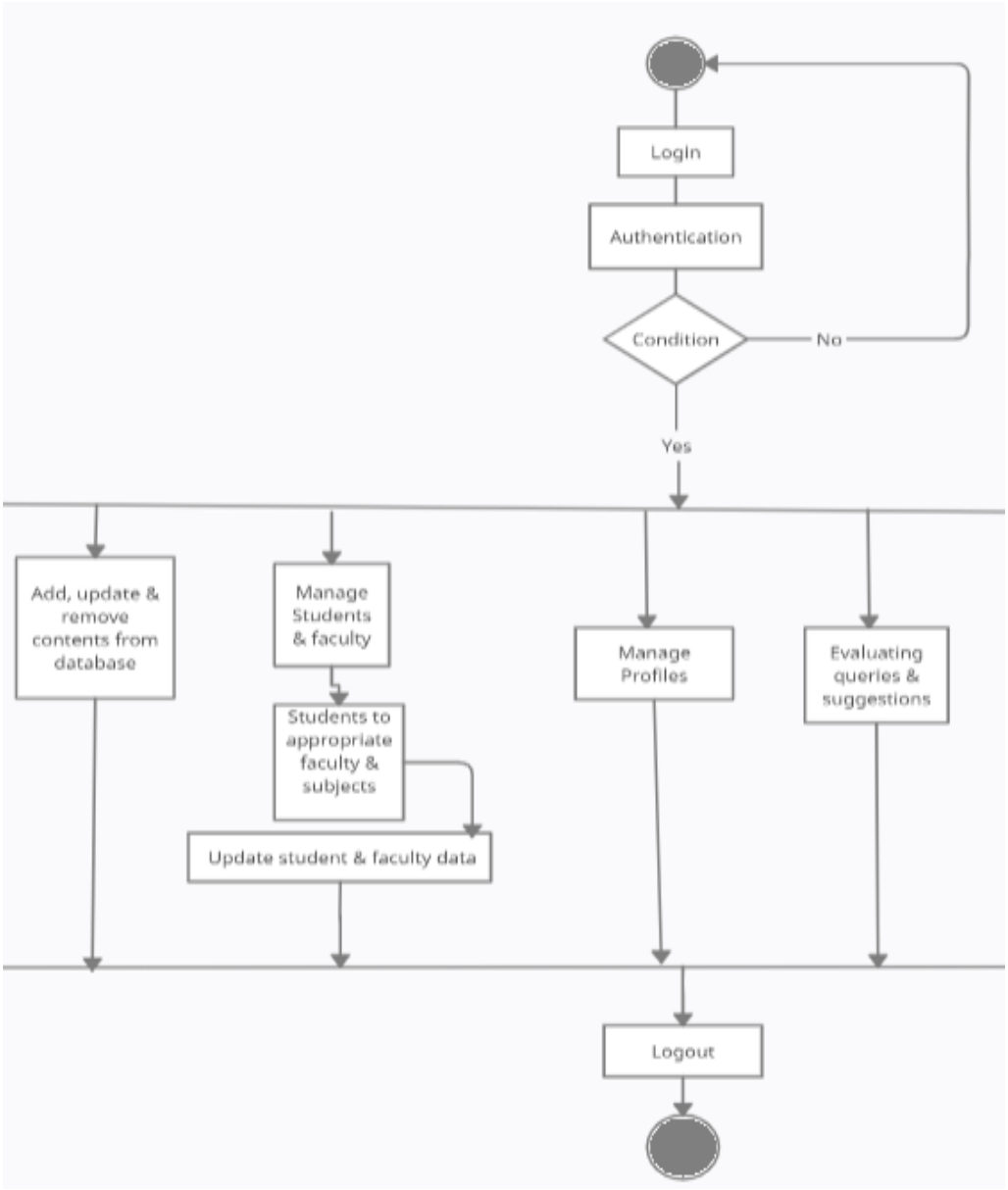


Figure 5.2: Activity diagram of admin

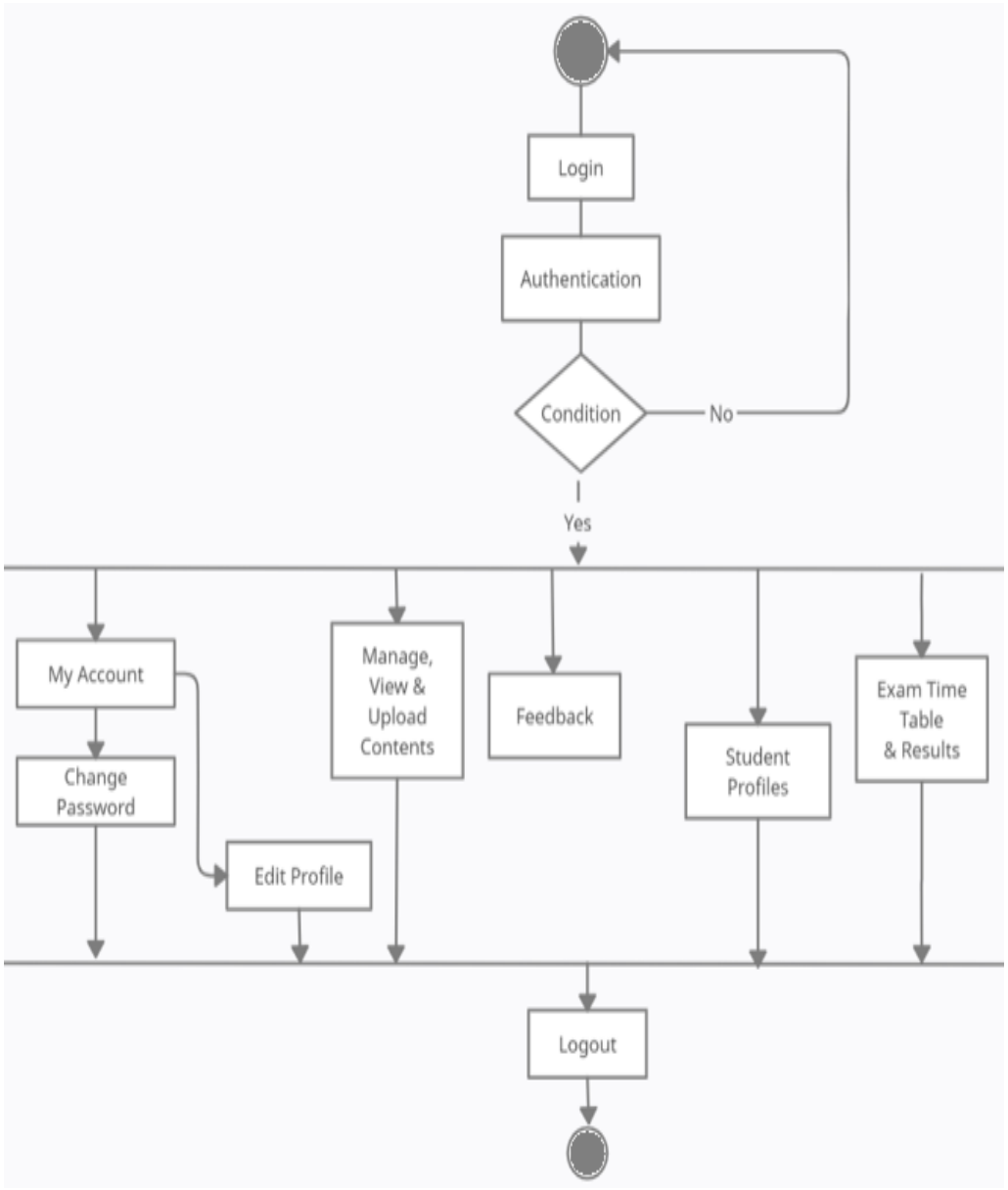


Figure 5.3: Activity diagram of Faculty

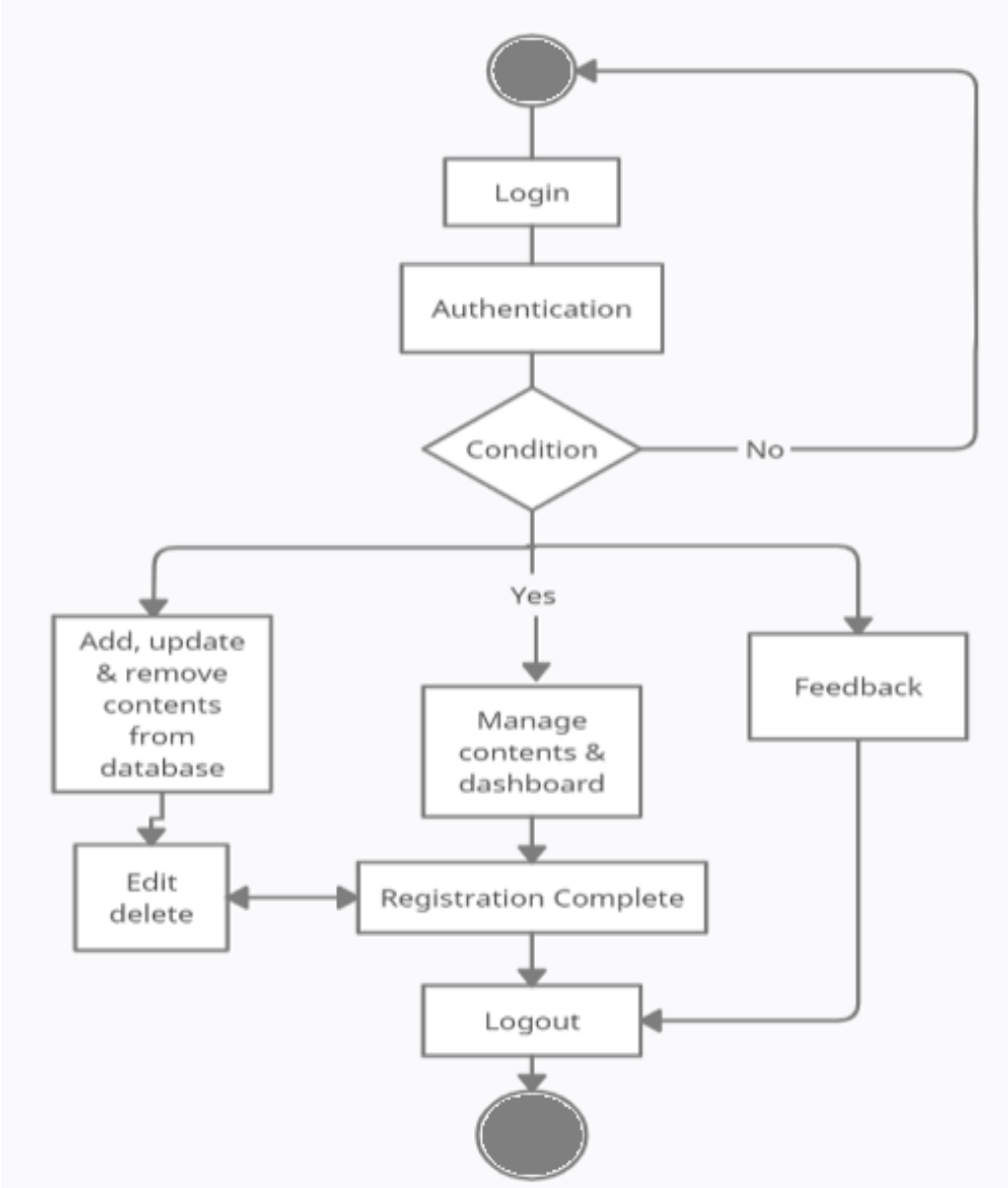


Figure 5.4: Activity diagram of Student

5.3 Sequence Diagram

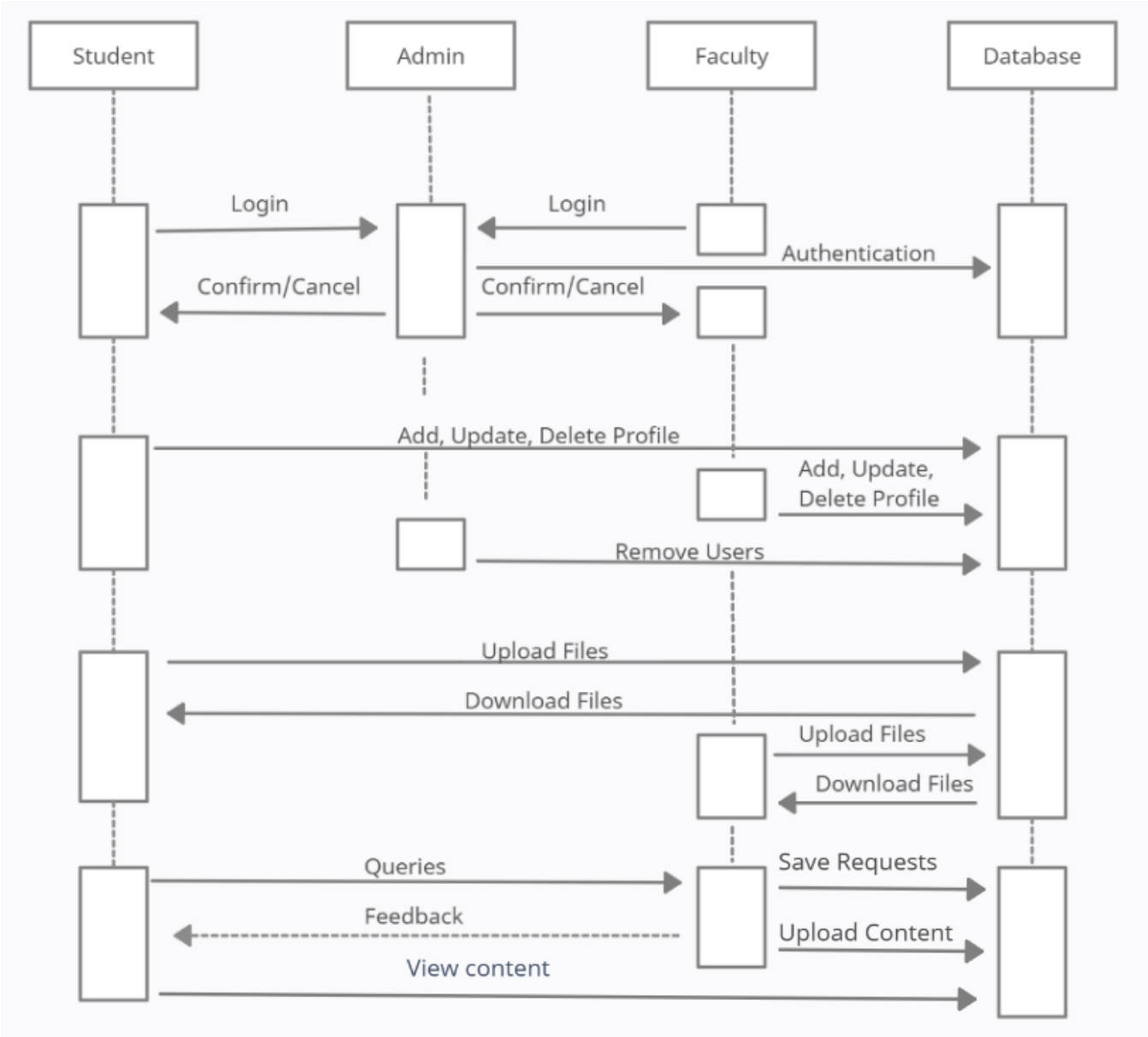


Figure 5.5: Sequence diagram

CHAPTER 6

DATA FLOW DIAGRAMS

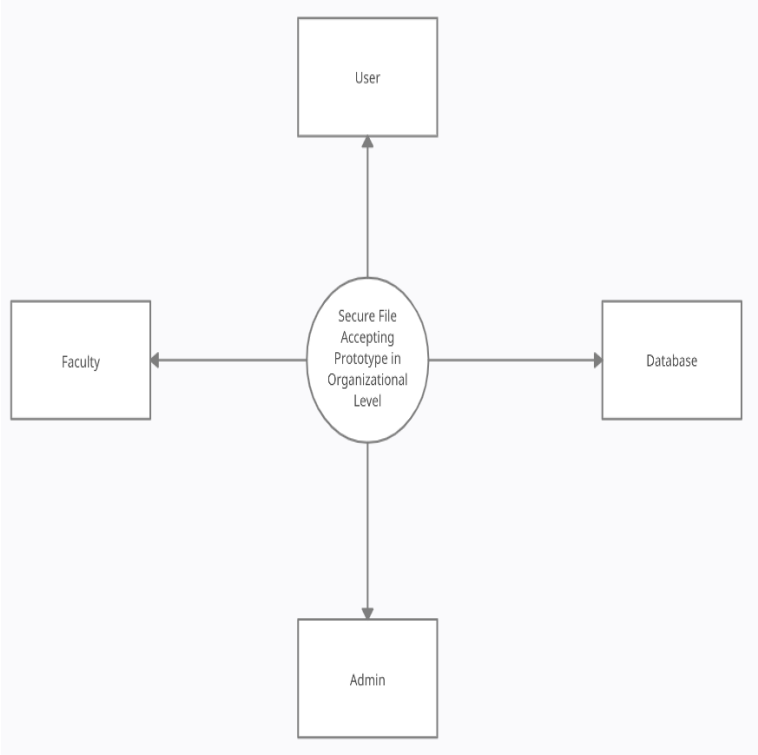


Figure 6.1: Data flow diagram - Level 0

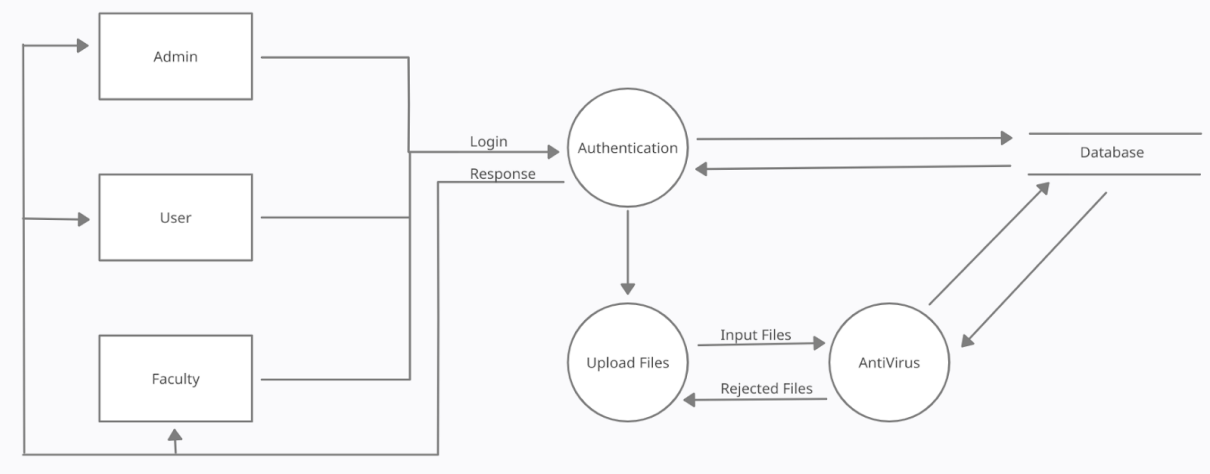


Figure 6.2: Data flow diagram - Level 1

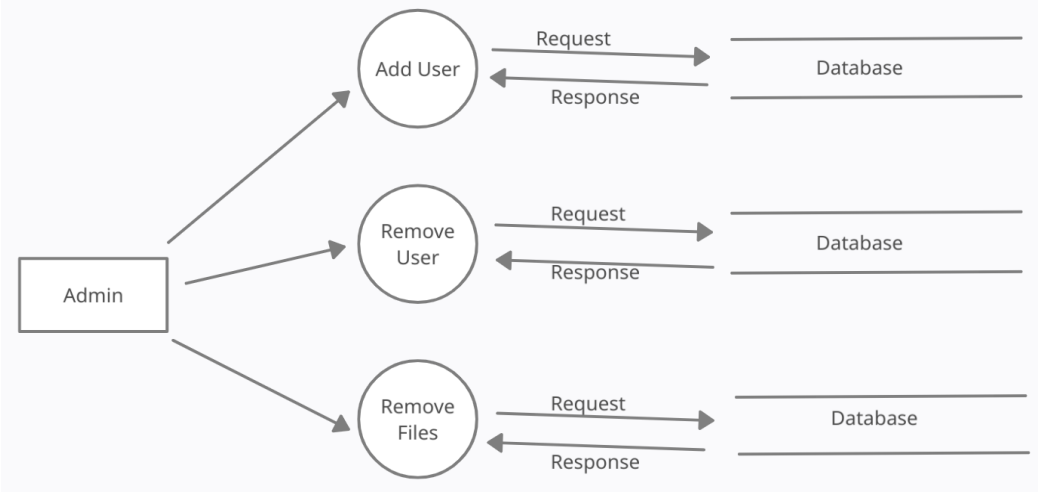


Figure 6.3: Data flow diagram - Level 2 - Admin

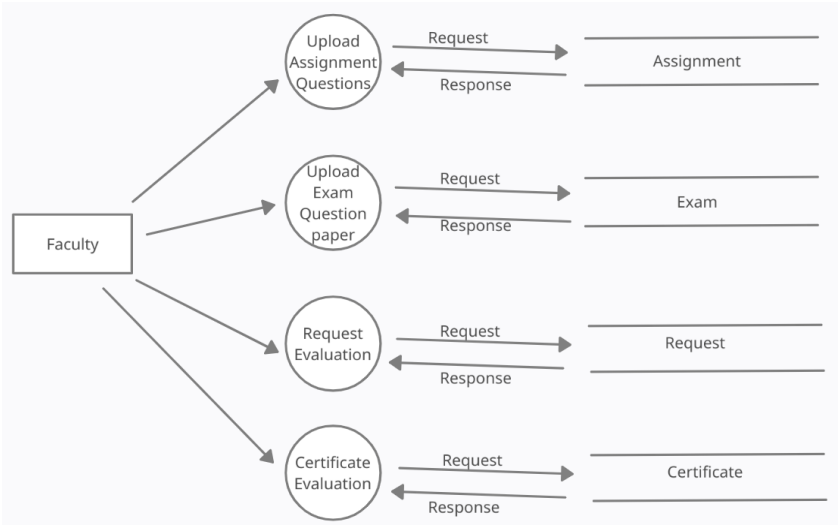


Figure 6.4: Data flow diagram - Level 2 - Faculty

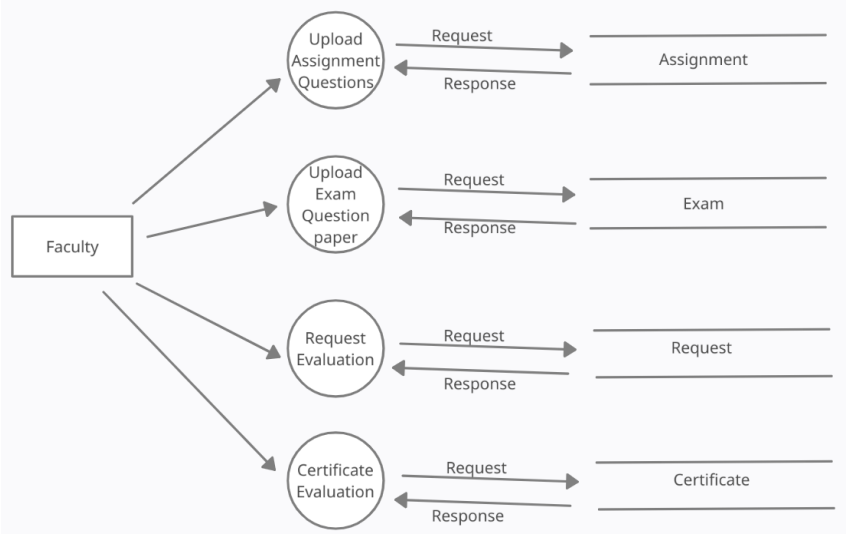


Figure 6.5: Data flow diagram - Level 2 - User (Student)

CHAPTER 7

CONCLUSION

A malware attack is a common cyber-attack where malware executes unauthorized actions on the victim's system. Antivirus manages with this malware's for a long term. When it comes to a long term the cost for maintain the antivirus for an organisation is very high. The 3- tier architecture that the websites are made has its own limitation that the files are scanned only after entering the application server. Thus we includes an additional layer of security by adding a separate sever for the antivirus program and completely apart from the other servers. The Antivirus stands for checking the hash values and continually checks the API's and extensions.

CHAPTER 8

REFERENCE

1. Study on application model of three-tiered architecture, Jun Tie, Jia Jin, Xiaorong WangSchool of Computer Science and Technology, Huazhong University of Science and Technology - College of Computer Science, South-Central University for Nationalities Wuhan, 430074, P.R. China IEEE 2011

2. A Comprehensive Review on Malware Detection Approaches - ÖMER ASLAN AND REFIK SAMET , (Member, IEEE) Computer Engineering Department, Ankara University, 06830 Ankara, Turkey Computer Engineering Department, Siirt University, 56100 Siirt, Turkey IEEE 2019

3. A Study on Web Application Security and Detecting Security Vulnerabilities - Sandeep Kumar, Renuka Mahajan, Naresh Kumar, Sunil Kumar khatri. IEEE 2017

4. Memory-Based Hardware Architectures to Detect ClamAV Virus Signatures with Restricted Regular Expression Features. - Nga Lam Or, Xing Wang, and Derek Pao Member IEEE 2016

5. Investigation and analysis on malware on websites - Takeshi Yagi,Naoto Tanimoto,Takeo Hariu and Mitsutaka Itoh NTT Information Sharing Platform Laboratories,NTT Cooperation. IEEE 2016

6. Malware detection and classification based on extraction of API sequences - Dolly Uppal, Rakhi Sinha, Vishakha Mehra and Vinesh Jain Department of Computer Engineering and Information Technology Government Engineering College Ajmer, India IEEE 2016

7. Multi-Level Security of Website: A Review of Use Multitier System with Many Password Encryption Method - Hind Saleem Ibrahim Harba IEEE 2015

8. Memory-efficient signature matching for ClamAV on FPGA - Tran Ngoc Thinh and Tran Trung Hieu Faculty of Computer Science and Engineering HCMC University of Technology Ho Chi Minh City, Vietnam IEEE 2014

9. String Searching Engine for Virus Scanning - Derek Pao, Xing Wang, Xiaoran Wang, Cong Cao, and Yuesheng Zhu, Senior Member, IEEE 2011

10. Content Management System : Comparative Case Study - Manish Nath , Anuja Arora Department of Computer Science and Information Technology, Jaypee Institute of Information Technology, Uttar Pradesh, INDIA IEEE 2010

11. Solution Architecture for N- Tier Applications - Toncy C Shan, Winnie W Hua IEEE 2006

12. Modeling 3-Tiered Web Applications - Xue Liu, Jin Heo and Lui Sha Department of Computer Science,University of Illinois at Urbana-Champaign IEEE 2005