Tool Assessment Report — Team 2

Sean Yap Yu Rong, Lim Ding Heng, Toh Yunqi Cheryl, Sun Shengran
IFS4205 Information Security Capstone Project, AY19/20, Semester 1 August 30, 2019

Overview

To understand our assessment of tools, it is crucial to first understand our system in mind. The system that we would like to call "NUSMed" is designed to make use of smartphones, as a reader, to read tokens in order to authenticate users into the web application. Therefore, this project will consist of two development projects with two separate source code repositories; one to facilitate the web application and another to facilitate the mobile application.

Largely, in order to facilitate the development of this huge project with only 4 team members, we have decided to proceed to run on top of the "WISA" solution stack. Many of the tools and software that we have chosen were influenced by the choice of using Windows Servers. This report will attempt to explain the decision to go with the Windows instead of Linux route and, in summary, explain why we settled on such choices despite other alternatives available currently in the market.

1 Frameworks and Languages

1.1 Web Application Framework and "Main Programming Language"

In this project, as according to the project scope, the Web application is the most important component, as such, the tools and software relevant to it was decided first. Considering the only experience within the team in web application development is related to .NET, only .NET web application frameworks were assessed; there were huge considerations that utilizing a framework that everyone had no experience on would be too risky due to the tight project timeline and lack of manpower. Additionally, there will not be any use of client-side frameworks, not that there is a need to create fanciful animations and etc. The only client-side framework we will be utilizing is Twitter Bootstrap; to aid page design, look and feel.

Below, are the frameworks that were assessed. They are all ASP.NET related frameworks but differ in the sense that they utilize different sets of runtimes, class libraries and view engines. They all allow for implementation for both Web Applications and Web API in a single project. ASP.NET adheres to the project requirements of being open-source.

- ASP.NET Core MVC 2.1.1, C#, Razor, with Entity, Identity and Authorization.

  ASP.NET Core MVC is a web application framework developed by Microsoft, which implements the model-view-controller (MVC) pattern. It is open-source and is bundled with "Identity" and "Membership" to provide mechanisms for authentication, role based access control and etc. It runs on the software framework called ".NET Core" that is both cross platform and open-source. Hence, the advantage is its ability to run cross platform naively and or via utilizing docker. It is to be noted that this "new" software framework has dropped support for Web Forms, another web application framework; more about this is discussed below.

  Entity, Identity and Authorization framework is mandatory to be used together in order to implement authentication, authorization and role access control. It is tied to Entity Framework, forcing developers to work with data in the form of domain-specific objects and properties. Additionally, the native way for interacting with data is Language Integrated Query (LINQ). While all these implementations would be certainly advantages in the long-run in multiple ways such as providing flexibility, code efficiency and great abstraction, it also means that there are multiple concepts that would prove too much to handle for such a short timeline. For example, every attribute in the database has to be scaffold-ed into classes before any development can start; a mandatory process that would take a week at least; not to mention, any changes to the database design would result in changes to the scaffold .

- ASP.NET MVC 5.2.7, C#, Razor, with Form Authentication and Membership.

  ASP.NET MVC 5.2.7 is an older version of the item discussed before. This older version instead runs

on the software framework, .NET Framework (implicit in the name as it is without the "Core"). This older version allows the use of simpler methods of authentication and role access control that is Form Authentication and Membership. Whilst this means that the web application losses some functionality such as the ability to use external identity providers, backup to non-relational stores and etc; these are features that are not relevant to this project. However, there is still the issue of utilizing the MVC software architectural pattern and Razor view engine that are alien to three-fourths of team members.

- **Chosen**: ASP.NET Web Forms, C#, with Form Authentication and Membership.

  To remove the steep leaning curve, ASP.NET Web Forms instead of MVC was chosen for its Model-view-viewmodel (MVVM) software architectural pattern that all team members are familiar with, also it separates the graphical user interface (GUI), in this case web pages, from the business logic (AKA back-end logic). This allows for an accelerated development if properly planned for; making up for the lack of manpower. This pattern, however, does come with drawbacks, such as confusing code in large pages and data bindings that result in large memory consumption. Regardless, since it is not expected for any web page in the planned application to be absurdly large; also, the project would not have many UI components, this issue should not pose a problem. Another factor in the consideration is the Web Forms view engine that allows for a more traditional approach to rendering pages that does not require more than rudimentary experience and understanding of client-side technologies; it abstracts it away and thus takes care of some part of the development.

  On the topic of security, there is little that Web Forms, despite being an older framework, does worse at as compared to the other alternate frameworks listed above. This framework simply does things in a different method such as in the case of software architectural pattern and server-side rendering components and etc. These are feature sets that we opt to make use of to aid in software engineering, ease development and most importantly, lessen man hours to focus on implementing security features.

  It is notable that while ASP.NET is regarded as open-source, the single component of Web Form controls are proprietary. The version of ASP.NET Framework that we will be using is 4.8, the most updated and recommended version to utilize.

1.2 Multi-Factor Authentication Framework

As per project requirements, a multi-factor authentication (MFA) subsystem is to be implemented and the second factor is preferably based on hardware so that it can also facilitate the requirement of validating a data entry of a comatose patient. Available choices for the hardware would be Bluetooth Low Energy (BLE) tags or Near-field communication (NFC) tags. Our MFA subsystem will thus consist of passwords which the user knows and the hardware which the user has.

- Bluetooth Low Energy (BLE) tags

  BLE tags operates the same way as Bluetooth except that it is much more energy efficient due to its low power consumption. It is able to cover a range of up to 10 meters but that is unnecessary for our authentication system since a larger coverage would only increase the chances of attacks such as sniffing or replay attacks. Using BLE also requires a power source which makes it impractical for our system as there is a chance where operations are affected when the BLE tags run out of power.

- **Chosen**: Near-field communication (NFC) tags

  NFC tags typically covers a range of up to 10 centimeters which makes it much more apt for our authentication system due to the short range. There are passive NFC tags which are powered from an NFC reader which essentially means these tags can last forever and work as long as there is an NFC reader to power it. These passive NFC tags are cheap and it is also possible for these tags to be implanted under the skin. The NFC tags can also be read by smartphones with an in-built NFC reader and is thus more feasible and practical since a majority of the population owns a smartphone.

1.3 Mobile Application Framework

As we have chosen NFC technology as part of the MFA process, we will thus build a mobile application for it since smartphones can function as a NFC reader as well. As such, the phones will minimally require an in-built NFC reader. Choosing to build a mobile application for the MFA process also makes it convenient for potential users since the mobile application could integrate functions such as uploading patient data like photos and videos.

- Mobile application for most mobile devices with in-built NFC reader
- **Chosen**: Android application for Android phones (at least version 4.0.3) with in-built NFC reader

  We do not have phones running on different operating systems (OS) or the necessary development environment for applications on other OS (e.g. XCode). Hence, we have chosen to only develop the application on Android and make the assumption that all potential users will use an Android phone with an in-built NFC reader. We will also make the assumption that the Android phones are running on at least version 4.0.3 (IceCreamSandwich) so that the developed application will function on 100% of the devices.

2 Infrastructure

Below is the Architecture Diagram for the proposed infrastructure of the system. Note that red arrows show communications essential to operations while blue arrows show admin communications.

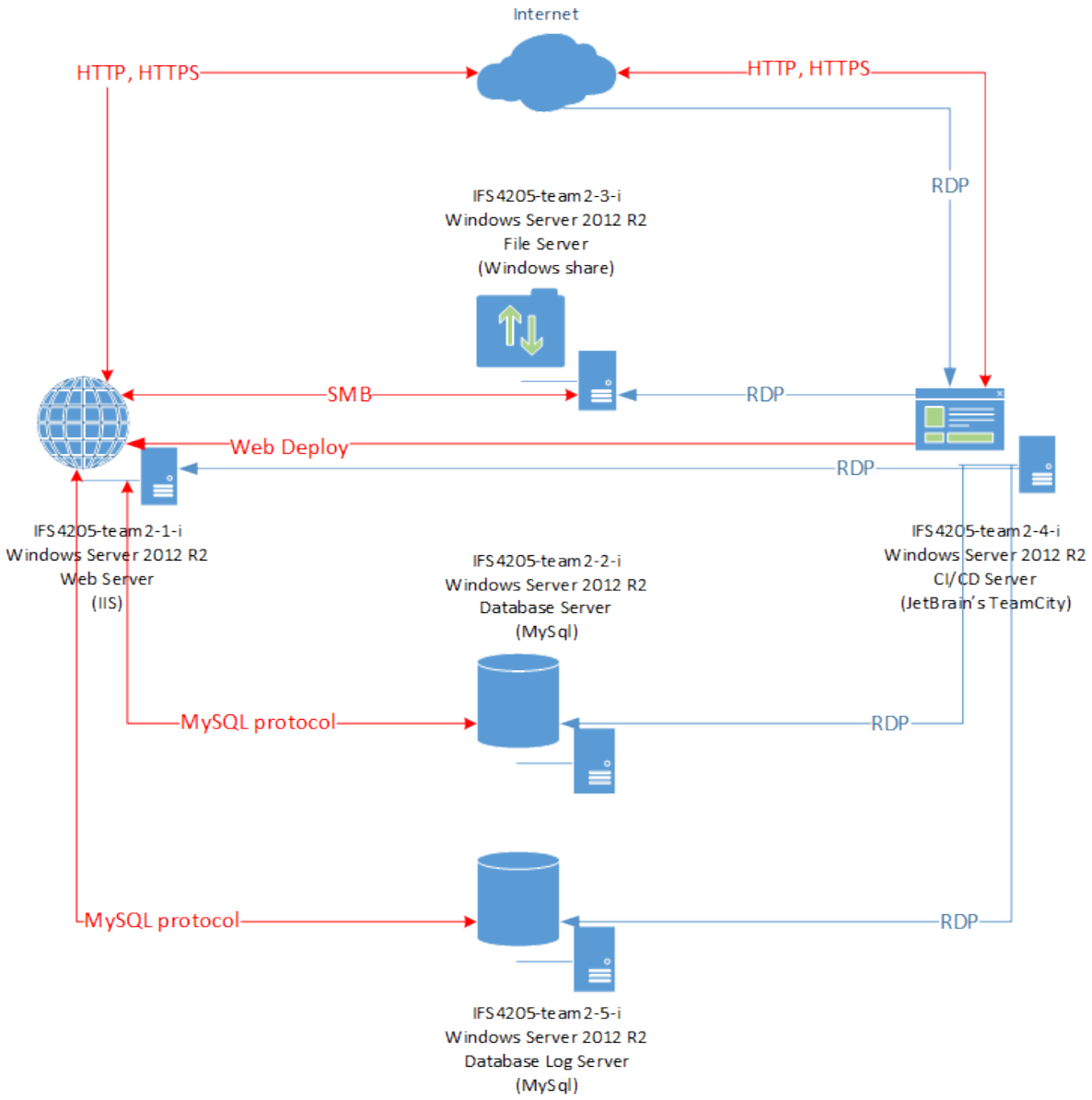

Figure 1: Architecture Diagram

2.1 Server Operating Systems

There were only two choices provided.

- Ubuntu 16.04.6 LTS

  Ubuntu is a free and open-source Linux distribution based on Debian. There were no qualms making use of this operating system. However, due to the web application framework selected, dockers or mono (the software project) has to be installed and configured to run the web application. While this additional step might be a small hassle, it does also come with performance issues.

- **Chosen**: Windows Server 2012 R2 Standard

  Windows Server 2012 R2 Standard was chosen as the operation system for all 5 servers in the infrastructure as firstly, it completes the solution stack resembling the "WINS" stack. This means that there are no additional software needed to support the system/application such that the solution of this system is at its minimal set, reducing the surface area of attack due to lesser and redundant application or services. Secondly, Windows Server 2012 R3 Standard comes in-built with Remote Desktop Protocol (RDP) that

provides a graphic interface for connecting to these Virtual Machines (VM); allowing for more efficient administrative control. Thirdly, Windows Server 2012 R3 Standard reduces the amount of third-party components needed to be installed manually from external sources; for example, Internet Information Services 10 (IIS) is in-built and File Server via Windows share or IIS File Server is in-built.

2.2 Web Software / Server

- **Chosen**: Internet Information Services (IIS) 10

  Due to the choice of running all the servers in the architecture on Windows Server 2012 R2 Standard, there are no alternatives besides using Internet Information Services (IIS) 10; not that it is an issue as it will allow easy configuration due to high compatibility with the chosen Web Application Framework.

2.3 Database Software / Server

The choice of running all servers on Windows Server 2012 R2 Standard does not affect our decision here as database servers tend to be operating system agnostic. However, as per project requirements, the databases do have to be a relational database management system (RDBMS) utilizing Structured Query Language (SQL). The items we had here are roughly equal to one another especially in terms of the workload and features that this project requires. Lastly, in this case, being open-source is an added benefit.

- Microsoft SQL Server (MSSQL) 2019

  Microsoft SQL Server is a relational database management system developed by Microsoft. It is not a database system that any team member has worked with. It is highly recommended to be used to conform to the Windows ecosystem; though not necessary in the case of this project as the features that MSSQL provides will not be used. However, the huge issue with MSSQL is the fact that it is not open-source and only its performance limited versions, express and developer versions, are free.

- PostgreSQL version 11.5

  PostgreSQL, also known as Postgres, is a relational database management system (RDBMS) emphasizing extensibility and technical standards compliance. It is a database system that is highly recommended for extensibility. It is also open-source. However, it does not include the feature of database modeling.

- **Chosen**: MySQL version 8

  MySQL was chosen for this project as checks all the boxes of being open-source, free and has the ability to perform database modeling. It is also the easiest and quickest database system to install and configure out of the alternatives.

2.4 Build Management and Continuous Integration (CI) Software / Server

As per project requirements, it is mandatory for the Build Management and Continuous Integration (CI/CD) Server to be on-premise instead of hosted. Additionally, due to the choice of running all the servers in the architecture on Windows Server 2012 R2 Standard, the chosen software to perform CI/CD has to be compatible on Windows. All of the software that we have shortlisted for assessment are equal to one another in terms of functionality.

- Azure DevOps Server Express 2019 (On-premise version), A.K.A. Team Foundation Server (TFS)

  Azure DevOps Server or Team Foundation Server (TFS) is a Microsoft product that, in summary, provides DevOps capabilities and more. It is tailored and integrated to work with Microsoft Visual Studio. Practically speaking, CI/CD software/servers is recommended to be chosen to fit user requirements as CI/CD software/server performance and capabilities are highly dependent on the project framework, type, language, task and etc. Hence, in the case of this project and its use of ASP.NET Framework to run the web application, the most practical solution would to make use of Azure DevOps Server to build, test and publish the web application. All these functionality, especially MSBuild are available out-of-the-box and little is required to be configured for implementation within the Windows ecosystem. However, there are concerns about utilizing Azure DevOps Server 2019, in that its latest version, the major upgrade to “Azure DevOps” is highly unstable and that its free version has a limitation of 5

users maximum. It is also a concern that it might be too feature rich, big and resource hungry to run on the provided VM's, given their allocated memory space.

- Jenkins

  Jenkins is an free and open-source automation server written in Java. It is a very popular CI/CD tool on the market today. It is feature rich, flexible and efficient. However, it relies on the concept of using plugins to perform rudimentary tasks such as building and publishing .NET web applications that we are targeting. Not to mention that to accommodate flexibility, it requires a large amount of both time and effort in configuration before use.

- **Chosen**: TeamCity

  TeamCity is a propriety build management and continuous integration server from JetBrains that was chosen for this project due to its value. It is famous for being reliable, high quality, easy to use, easy to navigate and, essentially, easy to configure; having provisions for popular development systems in-built, such as the ability to build and publish via MSBuild. It also has the ability to install plugins and increase its number of features. Most importantly, the free version of TeamCity has the generous restriction of a maximum of 100 build configurations and 1 build agent; more than enough for this project without worrying for expanding team size. Being said, TeamCity is notably a compromise of both Jenkins and Azure DevOps Server Express. Whilst it does not perform better in the niche areas of either Jenkins nor Azure DevOps Server Express, it does not have the drawbacks present in both alternatives too.

2.5 File Server

Due to the choice of running all the servers in the architecture on Windows Server 2012 R2 Standard, the chosen File Server has to be compatible on Windows.

- IIS File Server with Basic Authentication, File Transfer Protocol Over TLS (FTPS)

  FTPS is an extension to the commonly used File Transfer Protocol (FTP) that adds support for the Transport Layer Security (TLS). FTPS includes full support for the TLS and SSL cryptography protocols, including the use of server-side public key authentication certificates and client-side authorization certificates. It also supports compatible ciphers, including AES, RC4, RC2, Triple DES, and DES. It further supports hash functions SHA, MD5, MD4, and MD2.

  Historically FTP was not designed to be a secure protocol and there are numerous performance issues regarding the FTP protocol. For instance, it uses multiple connections to perform tasks, for example directory browsing and file transfers are done through separate connections.

- **Chosen**: Windows Shared Folder, Server Message Block (SMB) 3.0.2

  Server Message Block (SMB) is a network communication protocol for providing shared access to files, printers, and serial ports between nodes on a network. It works through a client-server approach, where a client makes specific requests and the server responds accordingly. It supports the same levels of security features as FTPS. However, utilizing Windows shared folder in conjunction with granting access to a specific service account for the web application to login to the folder would be a stronger authentication mechanism due to reliance on Windows Inter-process Communication (IPC) mechanisms to perform to facilitate communication between processes and computers over the shared nodes.

  Additionally, compared to FTPS, SMB is more efficient for the project needs as SMB, being a network communication protocol, allows for clients to randomly read or write from a file residing on a different node. FTPS on the other hand, requires the client to retrieve the entirety of the file. On the other hand, transfer speeds of FTPS tends to exceed SMB due to lesser overhead. Regardless, SMB 3.0.2 is chosen as it is more purposefully build for retrieving files for users over web applications whereby entire files need not be downloaded fully before other actions may take place. It is a more reliable protocol to communicate within a private network.

3 Project Management Tools

3.1 Source Code Management (SCM) and Issue Tracking System (ITS)

As per project requirements, it is mandatory for both source code management and issue management to be hosted in the cloud and for the hosting to allow project repositories to be initially private but subsequently public after development is completed. It is to be noted that all choices are roughly equal.

- Bitbucket Cloud

  Bitbucket Cloud is a web-based version control repository hosting service owned by Atlassian. Since it is mature in the industry, it is fully featured, supporting merge checks, wikis, issue tracking and more. It supports both Mercurial and Git and is free for students via enrollment using academic email addresses. It is noteworthy to point that though Bitbucket is popular overseas, it is rarely the first pick in Singapore and as such, no team member has any experience with Bitbucket.

- Gitlab

  GitLab is a web-based DevOps lifecycle tool that provides a Git-repository manager providing wiki, issue-tracking and CI/CD pipeline features, using an open-source license, developed by GitLab Inc. It has a free plan that provides unlimited private and public projects and collaborators. It is noteworthy that Gitlab is more CI/CD focused and whilst the free plan seems suitable and enticing, there are limits to the free plan that may prove troublesome. For example, the free plan does not allow for simple actions such as multiple approvers in code reviews, merge approvals and milestone lists and etc.

- Azure DevOps Services / Visual Studio Team Services (VSTS)

  Azure DevOps Services is a related product of Visual Studio. It is the implementation of the on-premises version of Team Foundation Server (TFS), also known as Azure DevOps Server; these details were discussed in the Infrastructure section of Build Management. It is functionality similar to Bitbucket Cloud, as detailed above. However, being part of the Visual Studio suite, it is well incorporated into Visual Studio's development process of ASP.NET projects. Though this is menial and no longer a huge feature as Git support has been incorporated into Visual Studio; giving the ability to interact with any and all Git repositories. Azure DevOps also support the use of Team Foundation Version Control (TFVC) as the source control. It's Basic plan is free of charge for up to five users.

  Azure DevOps Server certainly fits all the needs of this project and due it being purposefully developed for ASP.NET, it is the best choice in terms of ease of use and functionality wise. However, only one team member is familiar with it.

- **Chosen**: Github

  The chosen hosting service is Github with Git as the software versioning system. Github is a hosting service for software development version control using Git. Github is notable for providing fully featured access control and collaboration features such as bug tracking, feature requests, task management, wikis and etc. It provides a subset of features for free for all users and full functionality, besides organisation related features, to students that enroll into their educational program.

  The main reason to using Github is due to overall team experience in both using Github as a source-code hosting facility and using Git as a revision control system. It is also the recommended service to utilize by the supervisor of this project. Github has all the features necessary for this project without any drawbacks, such as Gitlab's lack of certain small functionalities. One issue with using Github is that organisation features are not entirely free and thus organisation repositories cannot be made private; which is a requirement. However, this can be circumvented by simply not utilizing organisations and creating separate repositories. This is a worthwhile mitigation to undertake in order to receive full project management features and utilize the existing experience and familiarity in the environment.

  An additional consideration is that Github is the largest host of source code which is helpful to the visibility of this project. This would be helpful in the future if the project were to become open-source.

4 Development Tools

Development tools were chosen based on the chosen specifications above, team experience and the recommendation by developer company. Hence, in this section, there are no alternatives to assess.

4.1 Integrated Development Environment (IDE) for C# Development

- **Chosen**: Visual Studio Community 2019

  Visual Studio Community 2019 will be used to develop the web application. It is the recommended IDE by Microsoft and is community supported. This community version is selected as it is free and supports the features that we require for the development of this project. Visual Studio invaluably includes inbuilt capabilities that would otherwise result in more work to include, as plugins, in other IDEs; for example, Git support, memory debugger, code publishing, code analysis, express server support and more.

4.2 Integrated Development Environment (IDE) for Android Application Development

- **Chosen**: Android Studio 3.5

  Android Studio will be used to develop an Android application which works with NFC. Android Studio is the official recommended development environment for Android developers. The application will also be developed with at least minimum API level 15 so that the application can function on 100% of Android phones running on version 4.0.3 (IceCreamSandwich) and above.

4.3 Integrated Development Environment (IDE) for SQL development, administration and database design

- **Chosen**: MySQL Workbench

  MySQL Workbench is the propriety SQL management tool provided by Oracle, developed by the community. It is provided free and comes bundled with MySQL community database server. It will not only be used as a database management tool but as a database design tool to model diagrams. It is to be noted that Visual Studio has an inbuilt database management tool but MySQL Workbench will be used for more highly intensive tasks.

5 Tools for Penetration Testing

The following are the tools that will be used to assess and provide insights to further secure the web application.

1. Vulnerability Scanners

   1.1. Nessus

   Nessus is one of the top proprietary vulnerability scanning tool which is able to detect misconfigurations or vulnerabilities on a server. In addition to its ease of use, it offers a Graphical User Interface, various types of report outputs and up to date information on new vulnerabilities and attacks. It is thus the ideal choice to assess our web application.

   1.2. Nikto

   Nikto is a free open-source command-line vulnerability scanner which can scan web servers for a large number of vulnerabilities. These vulnerabilities might include potentially dangerous files/CGIs, outdated server software or version specific problems for a server. It is one of the more popular and reputed web vulnerability scanner and will thus be used in tandem with Nessus.

2. Kali and related Tools

   2.1. Metasploit Framework

   The Metasploit Framework is an open-source modular penetration testing platform which contains a collection of modules which can aid with testing security vulnerabilities as well as executing

attacks. The huge collection of modules enables us to test the exploitability of a large number of vulnerabilities.

2.2. Nmap

Nmap is a free and well-known open-source tool which is able to aid with network discovery, port scanning and fingerprint operating systems. It would be able to discover if there are any open vulnerable ports on a server. It will be used in the penetration testing phase of the project due to its ease of use and the flexibility it provides from its diverse port scanning mechanisms.

2.3. Hydra

Hydra is a free popular brute force password cracking tool which can perform dictionary attacks against more than 50 protocols, of which includes common protocols such as telnet, ftp, http and https. It helps to test if any services are vulnerable due to a weak password.

2.4. SqlMap

SQLMap is an open-source tool which is essential to the penetration testing phase as the tool automates the process of detecting and exploiting SQL injection flaws. It is especially important to test for SQL injection flaws in a healthcare system due to the sensitive information stored in the databases.

2.5. Burp Suite

Burp Suite is a penetration testing tool for web applications. It is able to act as a proxy between the web browser and the web application to capture and analyze requests to and from the web application. That enables us to discover injection points in the web requests that might be exploitable.

2.6. Wireshark

Wireshark is a free open-source network traffic analyser that is crucial in helping us monitor the transmission of data between the mobile devices and the web server. This lets us ensure our designed communication protocol between the mobile devices and web server does not have any loopholes.

-End-