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

DETECTING FAKE APPLICATIONS IN SOCIAL NETWORK USING FRAPPE

[1] Polisetty Manoj Kumar, [2] P.Bhargavi,M.Tech
[1]M. Tech PG Scholar, [2]Asst.Professor
[1,2]Deptartment of CSE,
[1,2] SKR College of Engineering and Technology,Konduru satram (v),Manubolu(m),SPSR Nellore(dt),Andhra Pradesh.

Abstract: Online social networks (OSN) enable and encourage third party applications (apps) to enhance the user experience on these platforms. Such enhancements include interesting or entertaining ways of communicating among online friends, and diverse activities such as playing games or listening to songs For Example, With 20 million installs a day, third-party apps are a major reason for the popularity and addictiveness of Facebook. Unfortunately, hackers have realized the potential of using apps for spreading malware and spam. The problem is already significant, as we find that at least 13% of apps in our dataset are malicious. So far, the research community has focused on detecting malicious posts and campaigns. In this project, we ask the question: Given a Facebook application, can we determine if it is malicious? Our key contribution is in developing FRAPPE—Facebook's Rigorous Application Evaluator.

I. INTRODUCTION

1.1 What is networking?

Networking is the word basically relating to computers and their connectivity. It is very often used in the world of computers and their use in different connections. The term networking implies the link between two or more computers and their devices, with the vital purpose of sharing the data stored in the computers, with each other. The networks between the computing devices are very common these days due to the launch of various hardware and computer software which aid in making the activity much more convenient to build and use.

1.1.1 Characteristics of Networking:

The following characteristics should be considered in network design and ongoing maintenance:

1) Availability is typically measured in a percentage based on the number of minutes that exist in a year. Therefore, uptime would be the number of minutes the network is available divided by the number of minutes in a year.
2) Cost includes the cost of the network components, their installation, and their ongoing maintenance.
3) Reliability defines the reliability of the network components and the connectivity between them. Mean time between failures (MTBF) is commonly used to measure reliability.
4) .Securityincludes the protection of the network components and the data they contain and/or the data transmitted between them.
5) Speed includes how fast data is transmitted between network end points (the data rate).
6) Scalability defines how well the network can adapt to new growth, including new users, applications, and network components.
7) Topology describes the physical cabling layout and the logical way data moves between components

1.1.2 How networking works?

General Network Techniques

When computers communicate on a network, they send out data packets without knowing if anyone is listening. Computers in a network all have a connection to the network and that is called to be connected to a network bus. What one computer sends out will reach all the other computers on the local network.

Advantages of Networking:

1. Easy Communication:

It is very easy to communicate through a network. People can communicate efficiently usinga network with a group of people. They can enjoy the benefit of emails, instant messaging, telephony, video conferencing, chat rooms, etc.

2. Ability to Share Files, Data and Information:

This is one of the major advantages of networking computers. People can find and share information and data because of networking. This is beneficial for large organizations to maintain their data in an organized manner and facilitate access for desired people.

3. Sharing Hardware:

Another important advantage of networking is the ability to share hardware. For an example,a printer can be shared among the users in a network so that there's no need to have individual printers for each and every computer in the company. This will significantly reduce the cost of purchasing hardware.

Sharing Software:

Users can share software within the network easily. Networkable versions of software are available at considerable savings compared to individually licensed version of the same software. Therefore large companies can reduce the cost of buying software by networking their computers.

5. Security:

Sensitive files and programs on a network can be password protected. Then those files can only be accessed by the authorized users. This is another important advantage of networking when there are concerns about security issues. Also each and every user has their own set of privileges to prevent those accessing restricted files and programs.

6.Speed:

Sharing and transferring files within networks is very rapid, depending on the type ofnetwork. This will save time while maintaining the integrity of files.

Types of Networks:

Organizations of different structures, sizes, and budgets need different types of networks.Networks can be divided into one of two categories:

- peer-to-peer
- server-based networks

Peer-to-Peer Network:

A peer-to-peer network has no dedicated servers; instead, a number of workstations are connected together for the purpose of sharing information or devices. Peer-to-peer networks are designed to satisfy the networking needs of home networks or of small companies thatdo not want to spend a lot of money on a dedicated server but still want to have thecapability to share information or devices like in school, college, cyber cafe

Server-Based Networks:

In server-based network data files that will be used by all of the users are stored on the one server. With a server-based network, the network server stores a list of users who may use network resources and usually holds the resources as well.

This will help by giving you a central point to set up permissions on the data files, and it willgive you a central point from which to back up all of the data in case data loss should occur.

Network Communications:

- Computer networks use signals to transmit data, and protocols are the languages computers use to communicate.
- Protocols provide a variety of communications services to the computers on the network.
- Local area networks connect computers using a shared, half-duplex, baseband medium, and wide area networks link distant networks.
- Enterprise networks often consist of clients and servers on horizontal segments connectedby a common backbone, while peer-to-peer networks consist of a small number ofcomputers on a single LAN.

What Is A Social Network

social network service is a service which ―focuses on the building and verifying of online social networks for communities of people who share interests and activities, or who are interested in exploring the interests and activities of others, and which necessitates the use ofsoftware.‖

A report published by OCLC provides the following definition of social networking sites:

―Web sites primarily designed to facilitate interaction between users who share interests, attitudes and activities, such as Facebook, Mixi and MySpace.‖

How Canbe Social Networks used?

Social networks can provide a range of benefits to members of an organisation:

1. **Support for learning**: Social networks can enhance informal learning and support social connections within groups of learners and with those involved in the support of learning.

2. **Support for members of an organisation**: Social networks can potentially be used my all members of an organisation, and not just those involved in working with students. Social networks can help the development of communities of practice.

3. **Engaging with others**: Passive use of social networks can provide valuable business intelligence and feedback on institutional services (although this may give rise to ethical concerns).

4. **Ease of access to information and applications**: The ease of use of many social networking services can provide benefits to users by simplifying access to other tools andapplications. The Facebook Platform provides an example of how a social networking service can be used as an environment for other tools.

5. **Common interface**: A possible benefit of social networks may be the common interface which spans work / social boundaries. Since such services are often used in a personal capacity the interface and the way the service works may be familiar, thus minimising training and support needed to exploit the services in a professional context. This can, however, also be a barrier to those who wish to have strict boundaries between work and social activities.

Examples of Social Networking Services

Examples of popular social networking services include:

6. **Facebook**: Facebook is a social networking Web site that allows people tocommunicate with their friends and exchange information. In May 2007 Facebook launched the Facebook Platform which provides a framework for developers to create applications that interact with core Facebook features
7. **MySpace**: MySpace is a social networking Web site offering an interactive, user- submitted network of friends, personal profiles, blogs and groups, commonly used for sharing photos, music and videos..
8. **Ning**: An online platform for creating social Web sites and social networks aimed at users who want to create networks around specific interests or have limited technical skills.
9. **Twitter**: Twitter is an example of a micro-blogging service. Twitter can be used in a variety of ways including sharing brief information with users and providing support for one‘s peers.

Note that this brief list of popular social networking services omits popular social sharing services such as Flickr and YouTube.

Examples of Social Networking Services

The popularity and ease of use of social networking services have excited institutions with their potential in a variety of areas. However effective use of social networking services poses a number of challenges for institutions including long-term sustainability of the services; user concerns over use of social tools in a work or study context; a variety of technical issues and legal issues such as copyright, privacy, accessibility; etc.

Objective of the project:

We propose FRAPPEE for detecting fake application in a given social network. It Protect from anomaly and also virus files FRAPPE—a malicious app detector that utilizes our aggregation- based features in addition to the on-demand features.

EXISTING SYSTEM:

1. So far, the research community has paid little attention to OSN apps specifically. Most research related to spam and malware on Facebook has focused on detecting malicious posts and social spam campaigns.
2. Existing system works concentrated only on classifying individual URLs or posts as spam.
3. Existing system works focus on accounts created by spammers.
4. Gaoet al. analyzed posts on the walls of 3.5 million Facebook users and showed that 10% of links posted on Facebook walls are spam. They also presented techniques to identify compromised accounts and spam campaigns.
5. Yang et al. and Benevenutoet al. developed techniques to identify accounts of spammers on Twitter. Others have proposed a honey-pot-based approach to detect spam accounts on OSNs.

1.3 DISADVANTAGES OF EXISTING SYSTEM:

1. Existing system works concentrated only on classifying individual URLs or posts as spam, but not focused on identifying malicious applications that are the main source ofspam on Facebook.
2. Existing system works focused on accounts created by spammers instead of malicious application.
3. Existing system provided only a high-level overview about threats to the Facebook graph and do not provide any analysis of the system.

1.4 PROPOSED SYSTEM:

- In this paper, we develop FRAppE, a suite of efficient classification techniques for identifying whether an app is malicious or not. To build FRAppE, we use data from MyPage- Keeper, a security app in Facebook.
- We find that malicious applications significantly differ from benign applications withrespect to two classes of features: On-Demand Features and Aggregation-BasedFeatures.
- We present two variants of our malicious app classifier— FRAppE Lite and FRAppE.
- FRAppE Lite is a lightweight version that makes use of only the application features available on demand. Given a specific app ID, FRAppE Lite crawls the on-demand features for that application and evaluates the application based on these features in real time.
- FRAppE—a malicious app detector that utilizes our aggregation-based features in addition to the on-demand features.

1.5 ADVANTAGES OF PROPOSED SYSTEM:

1. The proposed work is arguably the first comprehensive study focusing on malicious Facebook apps that focuses on quantifying, profiling, and understanding malicious apps and synthesizes this information into an effective detection approach Several features used by FRAppE, such as the reputation of redirect URIs, the numberof required permissions, and the use of different client IDs in app installation URLs, are robust to the evolution of hackers.

2. LITERATURE SURVEY

2.1 A technique for computer detection and correction of spelling errors
AUTHORS: F. J. Damerau

The method described assumes that a word which cannot be found in a dictionary has at most one error, which might be a wrong, missing or extra letter or a single transposition. The unidentified input word is compared to the dictionary again, testing each time to see if the words match—assuming one of these errors occurred. During a test run on garbled text, correct identifications were made for over 95 percent of these error types.

2.2 LIBSVM: A library for support vector machines
AUTHORS: C.-C. Chang and C.-J. Lin

LIBSVM is a library for Support Vector Machines (SVMs). We have been actively developing this package since the year 2000. The goal is to help users to easily apply SVM to their applications. LIBSVM has gained wide popularity in machine learning and many other areas. In this article, we present all implementation details of LIBSVM. Issues such as solving SVM optimization problems theoretical convergence multiclass classification probability estimates and parameter selection are discussed in detail.

2.3 Beyond blacklists: Learning to detect malicious Websites from suspicious URL

AUTHORS: J. Ma, L. K. Saul, S. Savage, and G. M. Voelker
Malicious Web sites are a cornerstone of Internet criminal activities. As a result, there has been broad interest in developing systems to prevent the end user from visiting such sites. In this paper, we describe an approach to this problem based on automated URL classification, using statistical methods to discover the tell-tale lexical and host-based properties of malicious Web site URLs. These methods are able to learn highly predictive models by extracting and automatically analyzing tens of thousands of features potentially indicative of suspicious URLs. The resulting classifiers obtain 95-99% accuracy, detecting large numbers of malicious Web sites from their URLs, with only modest false positives.

2.4 Design and evaluation of a real-time URL spam filtering service

AUTHORS: K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song
On the heels of the widespread adoption of web services such as social networks and URL shorteners, scams, phishing, and malware have become regular threats. Despite extensive research, email-based spam filtering techniques generally fall short for protecting other web services. To better address this need, we present Monarch, a real-time system that crawls URLs as they are submitted to web services and determines whether the URLs direct to spam. We evaluate the viability of Monarch and the fundamental challenges that arise due to the diversity of web service spam. We show that Monarch can provide accurate, real-time protection, but that the underlying characteristics of spam do not generalize across web services. In particular, we find that spam targeting email qualitatively differs in significant ways from spam campaigns targeting Twitter. We explore the distinctions between email and Twitter spam, including the abuse of public web hosting and redirector services. Finally, we demonstrate Monarch's scalability, showing our system could protect a service such as Twitter--which needs to process 15 million URLs/day--for a bit under $800/day.

2.5 Detecting spammers on social networks

AUTHORS: G. Stringhini, C. Kruegel, and G.
Social networking has become a popular way for users to meet and interact online. Users spend a significant amount of time on popular social network platforms (such as Facebook, MySpace, or Twitter), storing and sharing a wealth of personal information. This information, as well as the possibility of contacting thousands of users, also attracts the interest of cybercriminals.
For example, cybercriminals might exploit the implicit trust relationships between users in order to lure victims to malicious websites. As another example, cybercriminals might find personal information valuable for identity theft or to drive targeted spam campaigns.
In this paper, we analyze to which extent spam has entered social networks. More precisely, we analyze how spammers who target social networking sites operate. To collect the data about spamming activity, we created a large and diverse set of "honey-profiles" on three large social networking sites, and logged the kind of contacts and messages that they received. We then analyzed the collected data and identified anomalous behavior of users who contacted our profiles. Based on the analysis of this behavior, we developed techniques to detect spammers in social networks, and we aggregated their messages in large spam campaigns. Our results show that it is possible to automatically identify the accounts used by spammers, and our analysis was used for take-down efforts in a real-world social network. More precisely, during this study, we collaborated with Twitter and correctly detected and deleted 15,857 spam profile.

3. SYSTEM STUDY AND ANALYSIS

3.1 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this syste

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

3.2 SOFTWARE REQUIREMENTS:

Operating system : Windows XP/7.
Coding Language : JAVA/J2EE
IDE : Net beans 7.4
Database : MYSQL

3.3 HARDWARE REQUIREMENTS:

System : i3 processor
Hard Disk : 40 GB.
Ram : 512 Mb.

3.4 USER REQUIREMENTS FUNCTIONAL REQUIREMENTS Admin

It is used by the admin for activating and deactivating user request

- User
- It is used to add document into database or used to download document from database.
- Modular Manager
- It is used to accept the document that is to be upload

3.5 NON-FUNCTIONAL REQUIREMENTS:

Non-Functional Requirements (quality attributes) ensure the delivery of an operable and manageable system which provides the required functionality reliable, uninterrupted or with minimal time of interruption even under unusual situations.

Security
Login requirements - access levels, CRUD levels
Password requirements - length, special characters, expiry, recycling policies Inactivity timeouts – durations, actions
Audit
Audited elements – what business elements will be audited? Audited fields – which data fields will be audited?
Audit file characteristics - before image, after image, user and time stamp, etc
Performance
Response times - application loading, screen open and refresh times, etcProcessing times – functions, calculations, imports, exports
Query and Reporting times – initial loads and subsequent loads
Capacity.
Availability
Hours of operation – when is it available? Consider weekends, holidays, maintenance times, etc
Locations of operation – where should it be available from, what are the connection requirements?
Reliability
Mean Time Between Failures – What is the acceptable threshold for down-time? e
.g. One a year, 4,000 hours

Mean Time To Recovery – if broken, how much time is available to get the system back up again?

Integrity

Fault trapping (I/O) – how to handle electronic interface failures, etc

Bad data trapping - data imports, flag-and-continue or stop the import policies, etc Data integrity – referential integrity in database tables and interfaces.Image compression and decompression standards

Recovery

Recovery process – how do recoveries work, what is the process? Recovery time scales – how quickly should a recovery take to perform?

Backup frequencies – how often is the transaction data, set-up data, and system (code) backed-up?

Backup generations - what are the requirements for restoring to previous instance(s)?

Compatibility

Compatibility on different operating systems – What does it have to be able to run on?

Compatibility on different platforms – What are the hardware platforms it needs to work on?

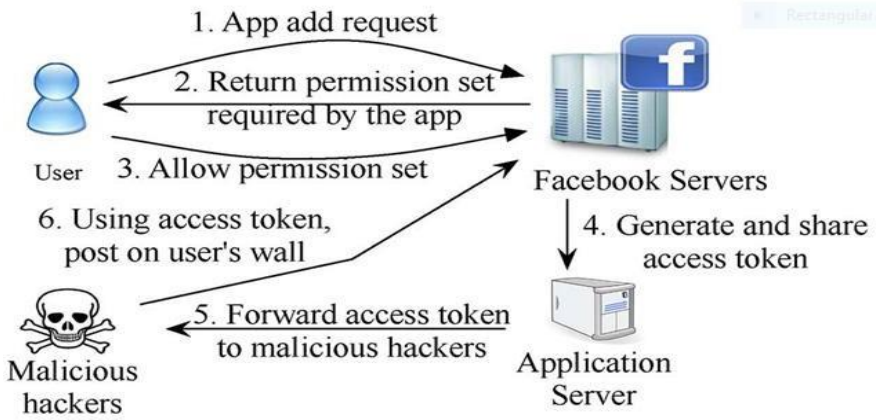
Maintainability

Conformance to architecture standards – What are the standards it needs to conform to or have exclusions from?

Conformance to design standards – What design standards must be adhered to or exclusions created?

4. SYSTEM DESIGN

ARCHITECTURE



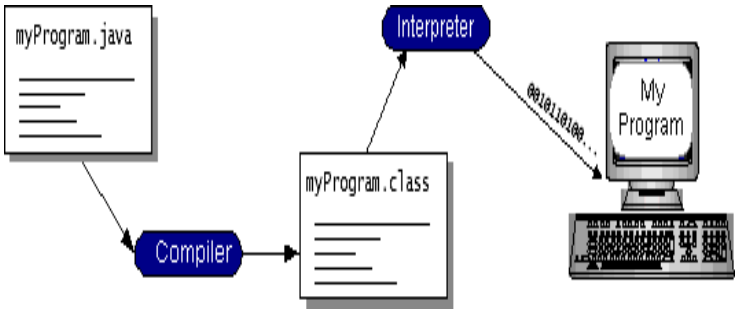
4.2 TECHNOLOGY DESCRIPTION

JavaTechnology

Java technology is both a programming language and a platformThe Java programming language is a high-level language that can be characterized by all of the following buzzwords:

- Simple
- Architecture neutral
- Object oriented
- Portable
- Distributed
- High performance
- Interpreted
- Multithreaded
- Robust
- Dynamic
- Secure

With most programming languages, you either compile or interpret a program so that youcan run it on your computer. The Java programming language is unusual in that a program isboth compiled and interpreted. With the compiler, first you translate a program into an intermediate language called Java byte codes —the platform-independent codes interpreted by the interpreter on the



Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.

You can think of Java byte codes as the machine code instructions for the **Java Virtual Machine** (Java VM). Every Java interpreter, whether it's a development tool or a Web browser that can run applets, is an implementation of the Java VM. Java byte codes help make ―write once, run anywhere‖ possible. You can compile your program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.

Why we use JAVA technology?

We can't promise you fame, fortune, or even a job if you learn the Java programming language. Still, it is likely to make your programs better and requires less effort than other languages. We believe that Java technology will help you do the following:

Get started quickly: Although the Java programming language is a powerful object-oriented language, it's easy to learn, especially for programmers already familiar with C or C++.

Write less code: Comparisons of program metrics (class counts, method counts, and so on) suggest that a program written in the Java programming language can be four times smaller than the same program in C++.

Write better code: The Java programming language encourages good coding practices, and its garbage collection helps you avoid memory leaks. Its object orientation, its JavaBeans component architecture, and its wide-ranging, easily extendible API let you reuse other people's tested code and introduce fewer bugs.

Develop programs more quickly: Your development time may be as much as twice as fast versus writing the same program in C++. Why? You write fewer lines of code and it is a simpler programming language than C++.

Avoid platform dependencies with 100% Pure Java: You can keep your program portable by avoiding the use of libraries written in other languages. The 100% Pure JavaTMProduct Certification Program has a repository of historical process manuals, white papers, brochures, and similar materials online.

Write once, run anywhere: Because 100% Pure Java programs are compiled into machine-independent byte codes, they run consistently on any Java platform.

Distribute software more easily: You can upgrade applets easily from a central server. Applets take advantage of the feature of allowing new classes to be loaded ―on the fly,‖ without recompiling the entire program.

ODBC

Microsoft Open Database Connectivity (ODBC) is a standard programming interface

for application developers and database systems providers. Before ODBC became a de facto standard for Windows programs to interface with database systems, programmers had to use proprietary languages for each database they wanted to connect to. Now, ODBC has made the choice of the database system almost irrelevant from a coding perspective, which is as it should be. Application developers have much more important things to worry about than the syntax that is needed to port their program from one database to another when business needs suddenly change.

Through the ODBC Administrator in Control Panel, you can specify the particular database that is associated with a data source that an ODBC application program is written to use. Think of an ODBC data source as a door with a name on it. Each door will lead you to a particular database. For example, the data source named Sales Figures might be a SQL Server database, whereas the Accounts Payable data source could refer to an Access database. The physical database referred to by a data source can reside anywhere on the LAN.

The ODBC system files are not installed on your system by Windows 95. Rather, they are installed when your setup a separate database application, such as SQL Server Client or Visual Basic 4.0. When the ODBC icon is installed in Control Panel, it uses a file called ODBCINST.DLL. It is also possible to administer your ODBC data sources through a stand-alone program called ODBCADM.EXE. There is a 16-bit and a 32-bit version of this program and each maintains a separate list of ODBC data sources.

From a programming perspective, the beauty of ODBC is that the application can be written to use the same set of function calls to interface with any data source, regardless of the database vendor. The source code of the application doesn't change whether it talks to Oracle or SQL Server. We only mention these two as an example. There are ODBC drivers available for several dozen popular database systems. Even Excel spreadsheets and plain text files can be turned into data sources. The operating system uses the Registry information written by ODBC Administrator to determine which low-level ODBC drivers are needed to talk to the data source (such as the interface to Oracle or SQL Server). The loading of the ODBC drivers is transparent to the ODBC application program. In a client/server environment, the ODBC API even handles many of the network issues for the application programmer.

The advantages of this scheme are so numerous that you are probably thinking there must be some catch. The only disadvantage of ODBC is that it isn't as efficient as talking directly to the native database interface. ODBC has had many detractors make the charge that it is too slow. Microsoft has always claimed that the critical factor in performance is the quality of the driver software that is used. In our humble opinion, this is true. The availability of good ODBC drivers has improved a great deal recently. And anyway, the criticism about performance is somewhat analogous to those who said that compilers would never match the speed of pure assembly language. Maybe not, but the compiler (or ODBC) gives you the opportunity to write cleaner programs, which means you finish sooner. Meanwhile, computers get faster every year.

JDBC

In an effort to set an independent database standard API for Java; Sun Microsystems developed Java Database Connectivity, or JDBC. JDBC offers a generic SQL database access mechanism that provides a consistent interface to a variety of RDBMSs. This consistent interface is achieved through the use of ―plug-in‖ database connectivity modules, or drivers. If a database vendor wishes to have JDBC support, he or she must provide the driver for each platform that the database and Java run on.

To gain a wider acceptance of JDBC, Sun basedJDBC's framework on ODBC. As you discovered earlier in this chapter, ODBC has widespread support on a variety of platforms. Basing JDBC on ODBC will allow vendors to bring JDBC drivers to market much faster than developing a completely new connectivity solution.

JDBC was announced in March of 1996. It was released for a 90 day public review that ended June 8, 1996. Because of user input, the final JDBC v1.0 specification was released soon after.

The remainder of this section will cover enough information about JDBC for you to know what it is about and how to use it effectively. This is by no means a complete overview of JDBC. That would fill an entire book.

JDBC Goals

Few software packages are designed without goals in mind. JDBC is one that, because of its many goals, drove the development of the API. These goals, in conjunction with early reviewer feedback, have finalized the JDBC class library into a solid framework for building database applications in Java.

4.3 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general- purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization,Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

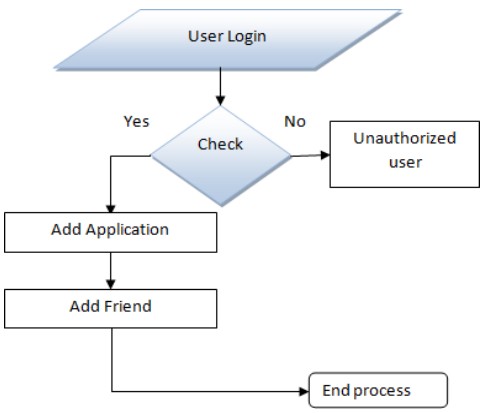
GOALS:

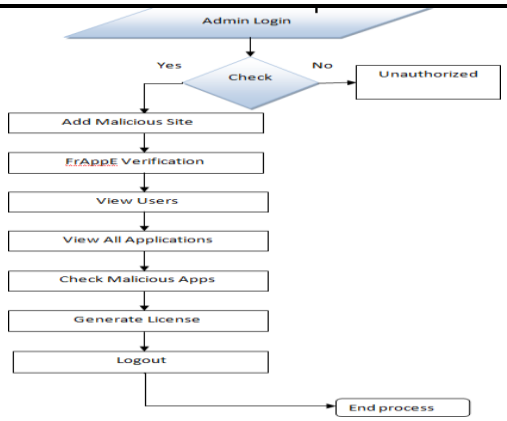
The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they candevelop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks,patterns and components.
7. Integrate best practices.

DATA FLOW DIAGRAM:

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can beused to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modifiedby a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at anylevel of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail





MODULES DESCRIPTION:

Data collection

The data collection component has two subcomponents: the collection of facebook apps withURLs and crawling for URL redirections. Whenever this component obtains a facebook app with a URL, it executes a crawling thread that follows all redirections of the URL and looks up the corresponding IP addresses. The crawling thread appends these retrieved URL and IP chains to the tweet information and pushes it into a queue. As we have seen, our crawler cannot reach malicious landing URLs when they use conditional redirections to evadecrawlers. However, because our detection system does not rely on the features of landing URLs, it works independently of such crawler evasions.

Feature extraction

The feature extraction component has three subcomponents: grouping of identical domains, finding entry point URLs, and extracting feature vectors.

To classify a post, MyPageKeeper evaluates every embedded URL in the post. Our key novelty lies in considering only the social context (e.g., the text message in the post, and the number of Likes on it) for the classification of the URL and the related post. Furthermore,we use the fact that we are observing more than one user, which can help us detect an epidemic spread.

It detects Presence of Spam keywords like _FREE‘, _DEAL‘ and _HURRY‘.

Training

The training component has two subcomponents: retrieval of account statuses and training ofthe classifier. Because we use an offline supervised learning algorithm, the feature vectors for training are relatively older than feature vectors for classification. To label the training vectors, we use the account status; URLs from suspended accounts are considered malicious whereas URLs from active accounts are considered benign. We periodically update our classifier using labeled training vectors.

Classification

The classification component executes our classifier using input feature vectors to classify suspicious URLs. When the classifier returns a number of malicious feature vectors, this component flags the corresponding URLs information as suspicious.

The classification module uses a Machine Learning classifier based on Support Vector Machines, but also utilizes several local and external white lists and blacklists that helpspeed up the process and increase the over-all accuracy. The classification module receives aURL and the related social context features extracted in the previous step.

These URLs, detected as suspicious, will be delivered to security experts or more sophisticated dynamic analysis environments for an in-depth investigation.

Detecting Suspicious

The Detecting Suspicious and notification module notifies all users who have social malware posts in their wall or news feed. The user can currently specify the notification mechanism, which can be a combination of emailing the user or posting a comment on the suspect posts.

V.SYSTEM TESTING

5.1 Implementation and Testing:

Implementation is one of the most importantphase in which one has to be cautions because all the efforts undertaken during the project will be very interactive. Implementation is the most crucial stage in achieving successful system and giving the users confidence that the new system is workable and effective. Each program is tested individually at the time of development using the sample data and has verified that these programs link together in the way specified in the program specification. The computer system and its environment are tested to the satisfaction of the user.

Implementation

The implementation phase is less creative than system design. It is primarily concerned with user training, and file conversion. The system may be requiring extensive user training. The initial parameters of the system should be modifying as a result of a programming. A simple operating procedure is provided so that the user can understand the different functionsclearly and quickly. The different reports can be obtained either on the inkjet or dot matrix printer, which is available at the disposal of the user.

The proposed system is very easyto implement. In general implementation is used to mean the process of converting a new or revised system design into an operational one.

5.2 Testing

Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property functions as a unit. The test data should be chosen such that it passed through all possible condition. Actually, testing is the state of implementation which aimed at ensuring that the system works accurately and efficiently before the actual operation commence. The following is the description of the testing strategies, which were carried out during the testing period.

System Testing

Testing has become an integral part of any system or project especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to user the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was testedlogically and pattern of execution of the program for a set of data are repeated. Thus, the code was exhaustively checked for all possible correct data and the outcomes were also checked.

Module Testing

To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus all the modules are individually tested from bottom up starting with the smallest and lowest modules andproceeding to the next level. Each module in the system is tested separately. For example thejob classification module is tested separately. This module is tested with different job and itsapproximate execution time and the result of the test is compared with the results that are prepared manually. The comparison shows that the results proposed system works efficiently than the existing system. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

Integration Testing

After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing results are very correct. Thus the mapping of jobs with resources is done correctly by the system.

Acceptance Testing

When that user fined no major problems with its accuracy, the system passers through a finalacceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation.

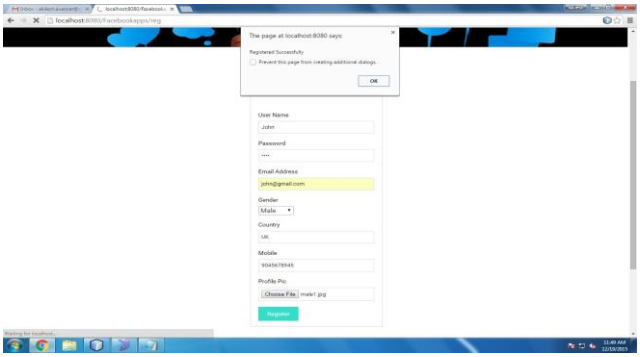
ALGORITHM

FRAppE as a step toward creating an independent watchdog for app assessment and ranking,so as to warn Facebook users before installing apps.

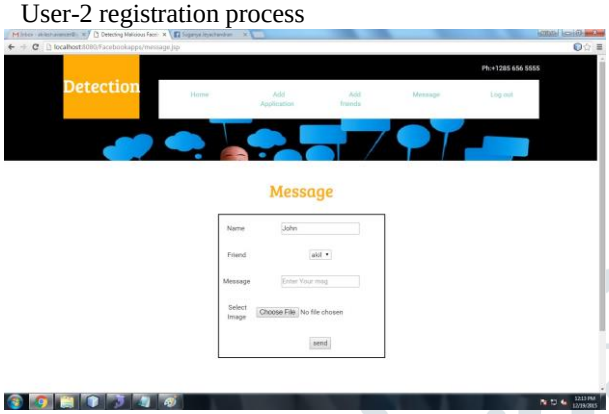
OUTPUT SCREENSHOTS

Home and user registration

Registration Successful



User-2 registration process



6. CONCLUSION

Applications present convenient means for hackers to spread malicious content on Facebook. However, little is understood about the characteristics of malicious apps and how they operate. In this paper, using a large corpus of malicious Facebook apps observed over a 5-month period, we showed that malicious apps differ significantly from benign apps with respect to several features. For example, malicious apps are Muchmore likely to share nameswith other apps, and they typically request fewer permissions than benign apps. Leveraging our observations, we developed FRAppE, an accurate classifier for detecting maliciousFacebook applications. Most interestingly, we highlighted the emergence of app-nets—largegroups of tightly connected applications that promote each other..

FUTURE ENHANCEMENT

We have concluded that the application focuses on detecting whether an applicationis malicious or benign. It helps to keep the user secured from third party applications andalso helps to hide bad posts, images and any other uploads. The work can be carried out on validation of a particular post, that is, for how long a post can be kept for a user to view. Theposts whether images, comments can be filtered as good or bad and can be avoided from sharing.We will continue to dig deeper into this ecosystem of malicious apps on Facebook, and we hope that Facebook will benefit from our recommendations for reducing the menace of hackers on their platform

1. P. Mell and T. Grance, ―The nist definition of cloud computing,‖ http://dx.doi.org/10.602/NIST.SP.800-145.
2. K. Ren, C. Wang, and Q. Wang, ―Security challenges for the public cloud,‖ IEEEInternet Computing, vol. 16, no. 1, pp. 69–73, 2012.
3. S. Kamara and K. Lauter, ―Cryptographic cloud storage,‖ in Springer RLCPS, January2010.
4. D. Song, D. Wagner, and A. Perrig, ―Practical techniques for searches on encrypteddata,‖ in IEEE Symposiumon Security and Privacy, vol. 8, 2000, pp. 44–55.
5. E.-J.Goh, ―Secure indexes,‖ IACR ePrint Cryptography Archive, http://eprint.iacr.org/2003/216, Tech. Rep., 2003.
6. D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, ―Public-key encryption withkeyword search,‖ in EUROCRYPR, 2004, pp. 506–522.
7. R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, ―Searchable symmetric encryption:improved deinitions and efficient constructions,‖ in ACM CCS, vol. 19, 2006, pp. 79–88.
8. M. Bellare, A. Boldyreva, and A. O‘Neill, ―Deterministic and efficiently searchableencryption,‖ in Springer CRYPTO, 2007