A NOVEL APPROACH TO RISK ASSESSMENT IN ONLINE SOCIAL NETWORKS

PUNEET PALAGI

ASSISTANT PROFESSOR

NEW HORIZON COLLEGE OF ENGINEERING

Abstract:

Although there is a dramatic increase in OSN usage – Facebook, for instance, has now 1.55 billion monthly active users, 1.31 billion mobile users, and 1.01 billion daily users1 there are also a lot of security/privacy concerns. One of the main source of these concerns is that OSN users establish new relationships with unknown people with the result of exposure of a huge amount of personal data. Unfortunately, very often users are not aware of this exposure as well as the serious consequences this might have.

Keywords: OSN, RISK ASSESSMENT, SOCIAL NETWORKS.

CHAPTER 1:

INTRODUCTION

DOMAIN SPECIFIC:

ONLINE Social Networks (OSNs) allow users to create a public or private profile, encourage sharing information and interests with other users and communicating with each other. As a result, OSNs are being used by millions of people and they are now part of our everyday life. People use OSNs to keep in touch with family, friends, and share personal information, as well as for business purposes. Users of an OSN build connections with their friends, colleagues and people over time. These connections form a social graph that controls how information spreads
in the social network.

Although there is a dramatic increase in OSN usage – Facebook, for instance, has now 1.55 billion monthly active users, 1.31 billion mobile users, and 1.01 billion daily users1 there are also a lot of security/privacy concerns. One of the main source of these concerns is that OSN users establish new relationships with unknown people with the result of exposure of a huge amount of personal data. Unfortunately, very often users are not aware of this exposure as well as the serious consequences this might have. Also, some users are less concerned about information privacy; therefore, they post more sensitive information on their profiles without specifying appropriate privacy settings and this can lead to security risks. As a result, today's social networks are exposed to many types of privacy and security attacks. These attacks exploit the OSN infrastructures to collect and expose personal information about their users, by, as an example, successfully convincing them to click on specific malicious links with

the aim of propagating these links in the network These attacks can either target users personal information as well as the personal information of their friends. Another widely used attack is the generation of fake profiles, which are generated with the sole purpose of spreading malicious content. In addition, there is a growing underground market on OSNs for malicious activities in that, for just a few cents, you can buy Facebook likes, share, Twitter followers, and fake accounts. Although many solutions, targeting one specific kind of attacks, have been recently proposed (see for instance), having a more general solution that can cope with the main privacy/security attacks that can be perpetrated using the social network graph is missing. In this paper, we make a step towards the definition of a unique tool that helps OSN providers as well as users to detect several types of attacks and, therefore, to have a global understanding of risky users in OSNs. We believe that the core of such a solution is a mechanism able to assign a risk score to each OSN account.

This risk estimation service will allow a user to make more consciously decisions about his/her privacy-risky activities within the network (e.g., answering to a friend request).Moreover, conducting a risk assessment in OSN will allow the service providers to minimize risk and help users to create and maintain a healthier friendship environment. We believe that a risk score can be useful for those users who want to 'inspect' their contacts, and also for the service providers wishing to know which users are risky. Therefore, our goal in this paper is to assign a risk score to each user, by taking into account both the user's activities and friendship patterns in the network. The goal is to compare the behavioral patterns of users with other users in the network to find anomalous behaviors. The key idea is that the more the user behavior diverges from what it can be considered as a 'normal behavior', the more it should be considered risky (i.e., with high risk score).

1.2 PROBLEM DEFINITION

One of the main source of these concerns is that OSN users establish new relationships with unknown people with the result of exposure of a huge amount of personal data. Unfortunately, very often users are not aware of this exposure as well as the serious consequences this might have. Also, some users are less concerned about information privacy; therefore, they post more sensitive information on their profiles without specifying appropriate privacy settings and this can lead to security risks. As a result, today's social networks are exposed to many types of privacy and security attacks. These attacks exploit the OSN infrastructures to collect and expose personal information about their users, by, as an example, successfully convincing them to click on specific malicious links with the aim of propagating these links in the network**.**

1.3 PROJECT PURPOSE

Our goal is to assign a risk score to each user, by taking into account both the user's activities and friendship patterns in the network. The goal is to compare the behavioral patterns of users with other users in the network to find anomalous behaviors. The key idea is that the more the user behavior diverges from what it can be considered as a 'normal behavior', the more it should be considered risky (i.e., with high risk score).

The first is the definition of a user behavioral profile able to catch those user's activities and interactions that are considered meaningful for the risk assessment. The second issue regards how to model a 'normal behavior'. In doing this, we have to consider that OSN population is really heterogeneous in observed behaviors. However, similar to real world, we expect that similar users (e.g., similar in activity level, gender, education, country, and so on) tend to follow similar rules (e.g., moral, social) with the results of similar behavioral models. Based on this principle, we propose a two-phase risk assessment, where users are first grouped together according to some

features meaningful for group identification. Then, for each identified group, we build one or more normal behavior models. To reach this goal, the key contributions include determining various user features to model normal/anomalous behaviors, and integrating them with probabilistic-clustering approach (the Expectation Maximization algorithm).

1.4 PROJECT FEATURES

A risk assessment based on the idea that the more a user behavior diverges from what it can be considered as a 'normal behavior', the more it should be considered risky. In doing this, we have takein into account that OSN population is really heterogeneous in observed behaviors. As such, it is not possible to define a unique standard behavioral model that fits all OSN users' behaviors. However, we expect that similar people tend to follow the similar rules with the results of similar behavioral models.

Group Identification Features

We recall that the aim of the first clustering is to group users for which similar behaviors are expected. At this purpose, group identification (GI) features should be those that are greatly discriminating, likes age, gender, but also those that

impact the possible users' behaviors, like, education and nationality. In addition to these features, we have to take into account that even if in the real world people with similar background usually behave in similar way, in an OSN this might be impacted by the users' attitude towards online social networks that might be different even for similar users.

Behavioral Features

In this section, we present the set of behavioral features (BF) on which our proposal is based. The design of this set is driven by the purpose of the system, that is, the detection of risky behaviors in OSN. At this aim, we have taken into account the behavioral patterns identified in the study of OSN attacks discussed in Section 2. This analysis brings us to the definition of 13 BFs, described in the following. As depicted in Table 1, this set is defined such to cover all attack patterns identified in Section 2. Another interesting quality of the identified set is that, as experiments in Section show every single feature is influent and relevant in

the risk assessment process. Before presenting the proposed behavioral features, we introduce two measures, used in features computation. The first is the user longevity, which is measured as the number of days since the user joined the OSN. The second one is the item longevity, where is measured as the number of days since an item has been uploaded in the OSN.

CHAPTER 2

LITERATURE SURVEY

Online and face-to-face discussions in the classroom: A study on the experiences of 'active' and 'silent' students

Even though the advantages of online discussions over face-to-face discussion formats has been extensively reported and investigated, the blending of online discussion tools in co-located classroom settings has been considered with far less intensity. In this paper, we report on secondary school students' experiences and preferences concerning two different discussion formats in co-located classroom settings, namely face-to-face (F2F) and synchronous, computer-mediated communication (CMC). In addition, we also differentiate between

students that are known to be active participants in F2F classroom discussions and those who usually remain silent. The findings highlight several advantages of CMC over F2F discussions in co-located settings and show that different students ('active' and 'silent') experience F2F and computer-mediated communication differently.

Graph-based Sybil Detection in social and information systems

Sybil attacks in social and information systems have serious security implications. Out of many defence schemes, Graph-based Sybil Detection (GSD) had the greatest attention by both academia and industry. Even though many GSD algorithms exist, there is no analytical framework to reason about their design, especially as they make different assumptions about the used adversary and graph models. In this paper, we bridge this knowledge gap and present a unified framework for systematic evaluation of GSD algorithms. We used this framework to show that GSD algorithms should be designed to find local community structures around known non-Sybil identities, while incrementally tracking changes in the graph as it evolves over time.

The Socialbot Network: When Bots Socialize for Fame and Money

Online Social Networks (OSNs) have become an integral part of today's Web. Politicians, celebrities, revolutionists, and others use OSNs as a podium to deliver their message to millions of active web users. Unfortunately, in the wrong hands, OSNs can be used to run astroturf campaigns to spread misinformation and propaganda. Such campaigns usually start off by infiltrating a targeted OSN on a large scale. In this paper, we evaluate how vulnerable OSNs are to a large-scale infiltration by socialbots: computer programs that control OSN accounts and mimic real users. We adopt a traditional web-based botnet design and built a Socialbot Network (SbN): a group of adaptive socialbots that are orchestrated in a command-and-control fashion. We operated such an SbN on Facebook—a 750 million user OSN—for about 8 weeks. We collected data related to users' behavior in response to a large-scale infiltration where socialbots were used to connect to a large number of Facebook users. Our results show that (1) OSNs, such as Facebook, can be infiltrated with a success rate of up to 80%, (2) depending on users' privacy settings, a successful infiltration can result in privacy breaches where even more users' data are exposed when compared to a purely public access, and (3) in practice, OSN security defenses, such as the Facebook Immune System, are not effective enough in detecting or stopping a large-scale infiltration as it occurs.

Scaling EM (Expectation Maximization) Clustering to Large Databases

Practical statistical clustering algorithms typically center upon an iterative refinement optimization procedure to compute a locally optimal clustering solution that maximizes the fit to data. These algorithms typically require many database scans to converge, and within each scan they require the access to every record in the data table. For large databases, the scans become prohibitively expensive. We present a scalable implementation of the Expectation-Maximization (EM) algorithm. The database community has focused on distance-based clustering schemes and methods have been developed to cluster either numerical or categorical data. Unlike distance-based algorithms (such as K-Means), EM constructs proper statistical models of the underlying data source and naturally generalizes to cluster databases containing both discrete-valued and continuous-valued data. The scalable method is based on a decomposition of the basic statistics the algorithm needs: identifying regions of the data that are compressible and regions that must be maintained in memory. The approach operates within the confines of a limited main memory buffer and requires at most a single database scan. Data resolution is

preserved to the extent possible based upon the size of the main memory buffer and the fit of the current clustering model to the data. We extend the method to efficiently update multiple models simultaneously. Computational tests indicate that this scalable scheme outperforms sampling-based approaches – the straightforward alternatives to "scaling" traditional in-memory implementations to large database

Future Work:

Social networking sites have been increasingly gaining popularity. Well-known sites such as Facebook have been reporting growth rates as high as 3% per week. Many social networking sites have millions of registered users who use these sites to share photographs, contact long-lost friends, establish new business contacts and to keep in touch. In this paper, we investigate how easy it would be for a potential attacker to launch automated crawling and identity theft attacks against a number of popular social networking sites in order to gain access to a large volume of personal user information. The first attack we present is the automated identity theft of existing user profiles and sending of friend requests to the contacts of the cloned victim. The hope, from the attacker's point of view, is that the contacted users simply trust and accept the friend request. By establishing a friendship relationship with the contacts of a victim, the attacker is able to access the sensitive personal information provided by them. In the second, more advanced attack we present, we show that it is effective and feasible to launch an automated, cross-site profile cloning attack. In this attack, we are able to automatically create a forged profile in a network where the victim is not registered yet and contact the victim's friends who are registered on both networks. Our experimental results with real users show that the automated attacks we present are effective and feasible in practice.

Requirements Specification

Use this Requirements Specification template to document the requirements for your product or service, including priority and approval. Tailor the specification to suit your project, organizing the applicable sections in a way that works best, and use the checklist to record the decisions about what is applicable and what isn't.

The format of the requirements depends on what works best for your project.

This document contains instructions and examples which are for the benefit of the person writing the document and should be removed before the document is finalized.

To regenerate the TOC, select all (CTL-A) and press F9.

Executive Summary

Project Overview

Describe this project or product and its intended audience, or provide a link or reference to the project charter.

Purpose and Scope of this Specification

Describe the purpose of this specification and its intended audience. Include a description of what is within the scope what is outside of the scope of these specifications. For example:

In scope

This document addresses requirements related to phase 2 of Project A:

modification of Classification Processing to meet legislative mandate ABC.

modification of Labor Relations Processing to meet legislative mandate ABC.

Out of Scope

The following items in phase 3 of Project A are out of scope:

modification of Classification Processing to meet legislative mandate XYZ.

modification of Labor Relations Processing to meet legislative mandate XYZ.

(Phase 3 will be considered in the development of the requirements for Phase 2, but the Phase 3 requirements will be documented separately.)

Product/Service Description

In this section, describe the general factors that affect the product and its requirements. This section should contain background information, not state specific requirements (provide the reasons why certain specific requirements are later specified).

Product Context

How does this product relate to other products? Is it independent and self-contained? Does it interface with a variety of related systems? Describe these relationships or use a diagram to show the major components of the larger system, interconnections, and external interfaces.

User Characteristics

Create general customer profiles for each type of user who will be using the product. Profiles should include:

Student/faculty/staff/other

experience

technical expertise

other general characteristics that may influence the product

Assumptions

List any assumptions that affect the requirements, for example, equipment availability, user expertise, etc. For example, a specific operating system is assumed to be available; if the operating system is not available, the Requirements Specification would then have to change accordingly.

Constraints

Describe any items that will constrain the design options, including

parallel operation with an old system

audit functions (audit trail, log files, etc.)

access, management and security

criticality of the application

system resource constraints (e.g., limits on disk space or other hardware limitations)

other design constraints (e.g., design or other standards, such as programming language or framework)

Dependencies

List dependencies that affect the requirements. Examples:

Error! Reference source not found.Error! Reference source not found., **Error! Reference source not found.**This new product will require a daily download of data from X,

Module X needs to be completed before this module can be built.

Requirements

Describe all system requirements in enough detail for designers to design a system satisfying the requirements and testers to verify that the system satisfies requirements.

Organize these requirements in a way that works best for your project. See for different ways to organize these requirements.

Describe every input into the system, every output from the system, and every function performed by the system in response to an input or in support of an output. (Specify what functions are to be performed on what data to produce what results at what location for whom.)

Each requirement should be numbered (or uniquely identifiable) and prioritized.

See the sample requirements in Functional Requirements, and **Error! Reference source not found.**, as well as these example priority definitions:

Priority Definitions

The following definitions are intended as a guideline to prioritize requirements.

Priority 1 – The requirement is a “must have” as outlined by policy/law

Priority 2 – The requirement is needed for improved processing, and the fulfillment of the requirement will create immediate benefits

Priority 3 – The requirement is a “nice to have” which may include new functionality

It may be helpful to phrase the requirement in terms of its priority, e.g., "The value of the employee status sent to DIS **must be** either A or I" or "It **would be nice** if the application warned the user that the expiration date was 3 business days away". Another approach would be to group requirements by priority category.

A good requirement is:

Correct

Unambiguous (all statements have exactly one interpretation)

Complete (where TBDs are absolutely necessary, document why the information is unknown, who is responsible for resolution, and the deadline)

Consistent

Ranked for importance and/or stability

Verifiable (avoid soft descriptions like “works well”, “is user friendly”; use concrete terms and specify measurable quantities)

Modifiable (evolve the Requirements Specification only via a formal change process, preserving a complete audit trail of changes)

Does not specify any particular design

Traceable (cross-reference with source documents and spawned documents).

Functional Requirements

In the example below, the requirement numbering has a scheme - BR_LR_0## (BR for Business Requirement, LR for Labor Relations). For small projects simply BR-## would suffice. Keep in mind that if no prefix is used, the traceability matrix may be difficult to create (e.g., no differentiation between '02' as a business requirement vs. a test case)

The following table is an example format for requirements. Choose whatever format works best for your project.

Req#	**Requirement**	**Comments**	**Priority**	**Date Rvwd**	**SME Reviewed / Approved**
BR_LR_05	The system should associate a supervisor indicator with each job class.	Business Process = "Maintenance	3	7/13/04	Bob Dylan, Mick Jagger
BR_LR_08	The system should handle any number of fees (existing and new) associated with unions.	Business Process = "Changing Dues in the System" An example of a new fee is an initiation fee.	2	7/13/04	Bob Dylan, Mick Jagger
BR_LR_10	The system should capture and maintain job class status (i.e., active or inactive)	Business Process = "Maintenance" Some job classes are old and are no longer used. However, they still need to be maintained for legal, contract and historical purposes.	2	7/13/04	Bob Dylan, Mick Jagger
BR_LR_16	The system should assign the Supervisor Code based on the value in the Job Class table and additional criteria as specified by the clients.	April 2005 – New requirement. It is one of three new requirements from BR_LR_03.	2		
BR_LR_18	The system should provide the Labor Relations office with the ability to override the system-derived Bargaining Unit code and the Union Code for to-be-determined employee types, including hourly appointments.	April 2005 – New requirement. It is one of three new requirements from BR_LR_04. 5/11/2005 – Priority changed from 2 to 3.	2 3		

CHAPTER 3

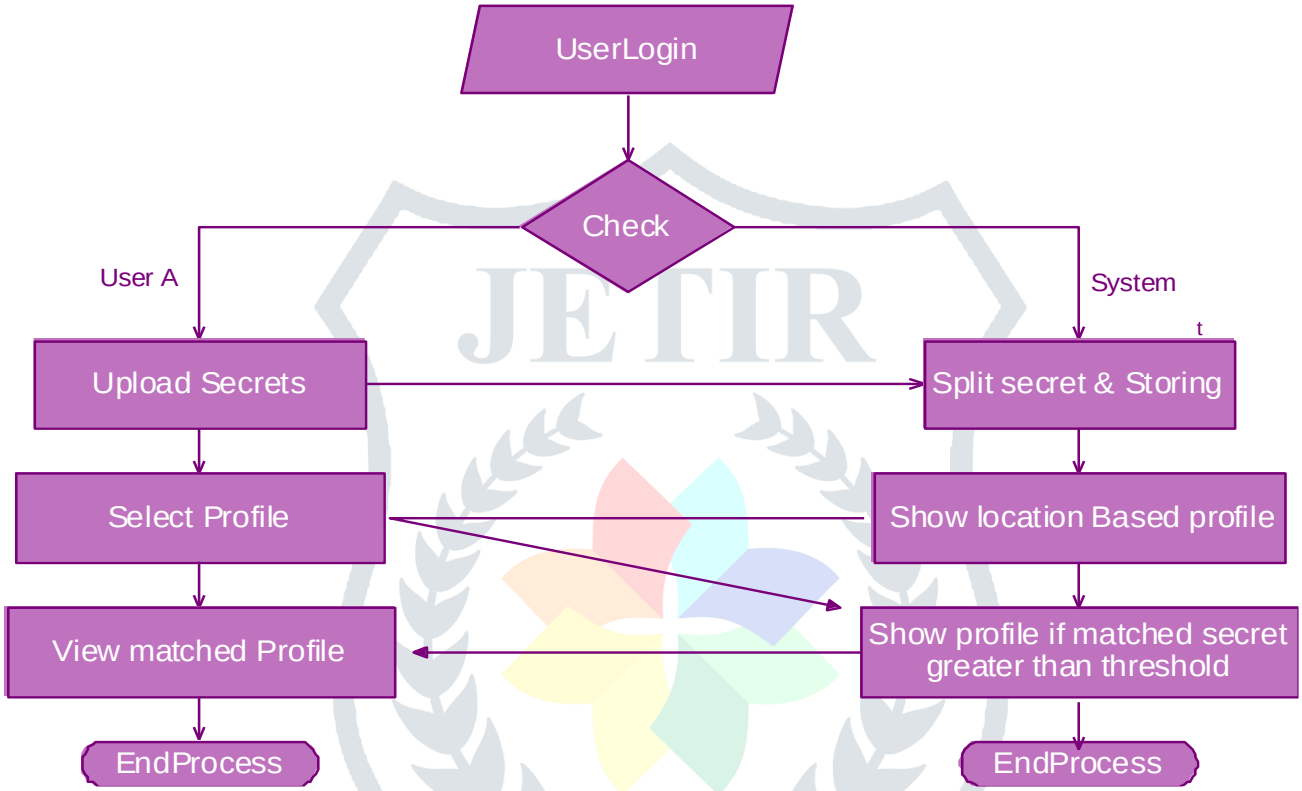
DESIGN

Data Flow Diagram / Use Case Diagram / Flow Diagram

The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of the input data to the system, various processing carried out on these data, and the output data is generated by the system.

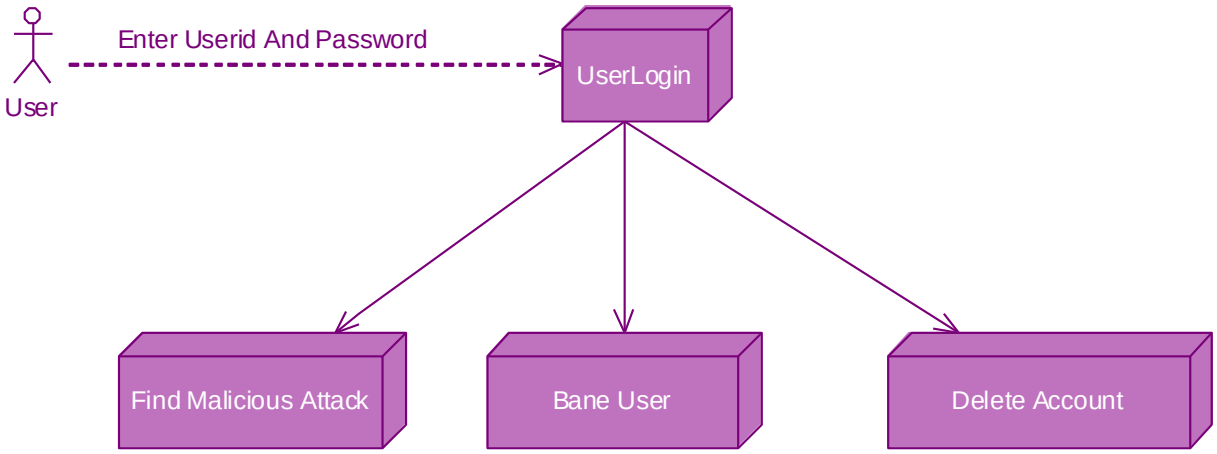
3.1 Data Flow Diagram:

(User):



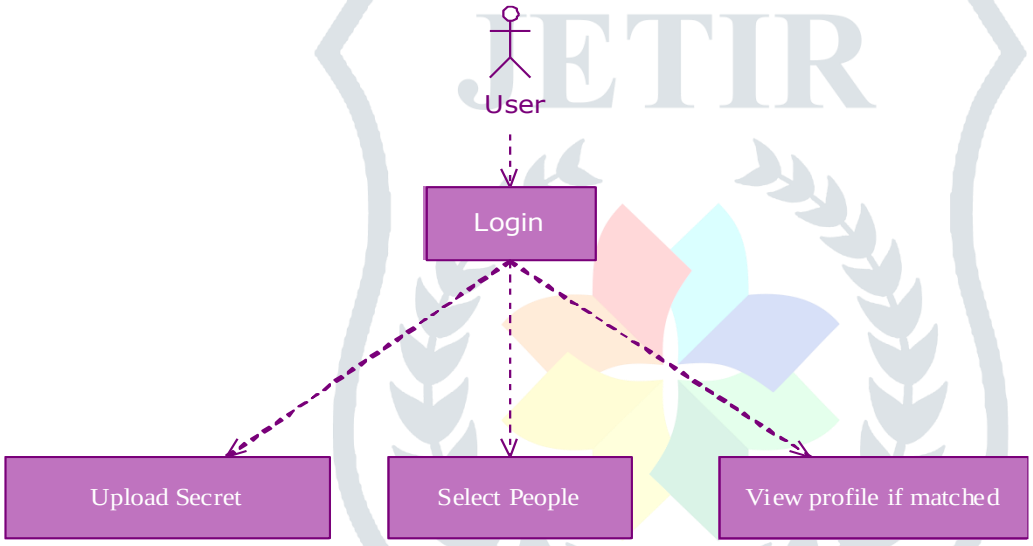
3.2 Component Diagram:

User:



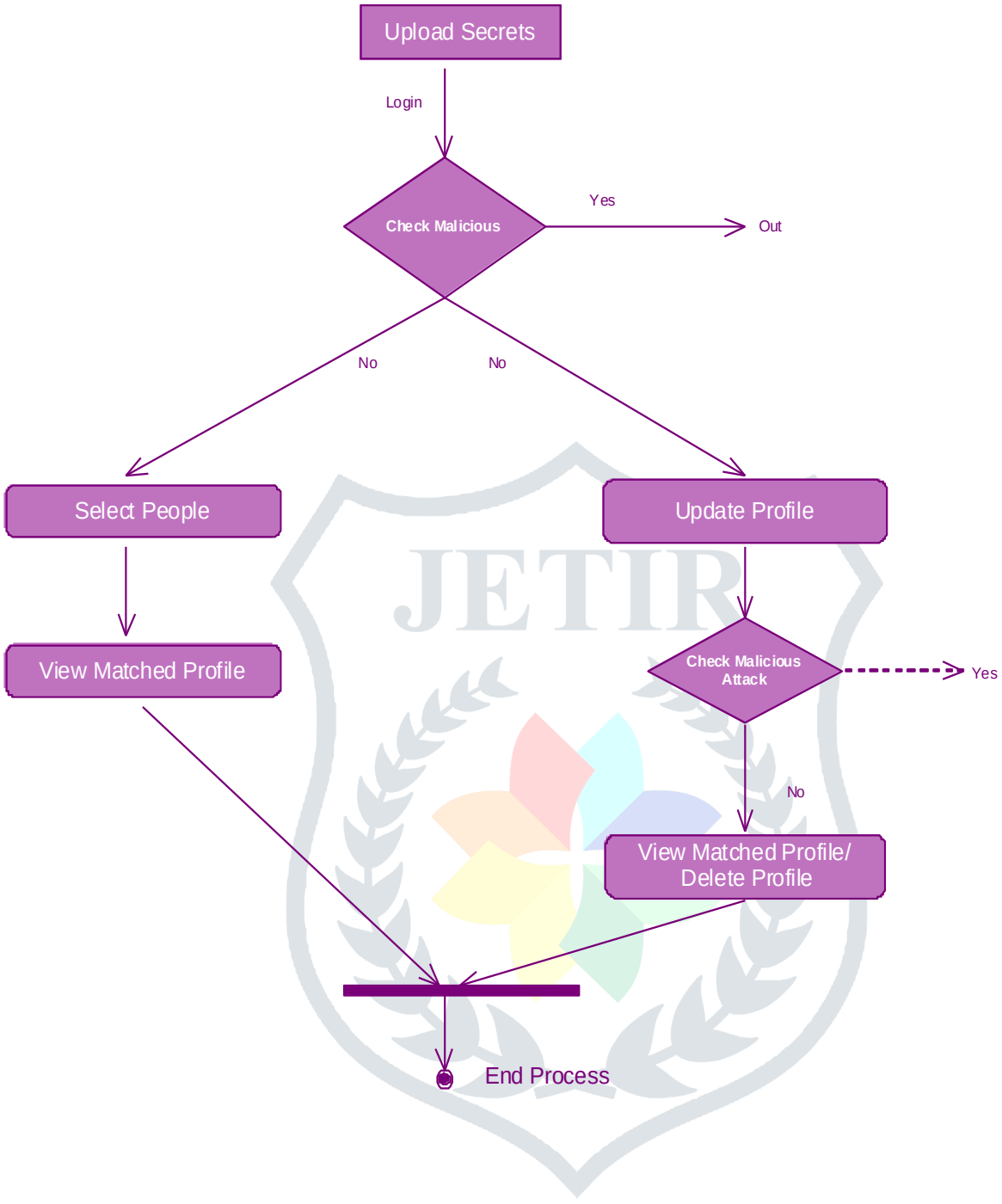
3.3 Use case Diagram:

(User):



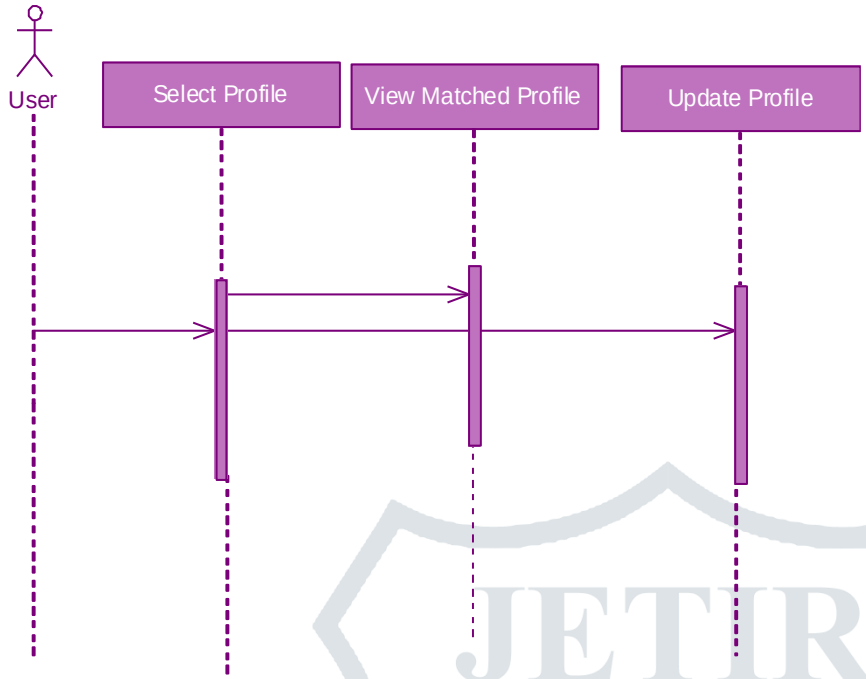
3.4 ACTIVITY DIAGRAM:

(User)



3.5 Sequence Diagram:

(User):



CONCLUSION

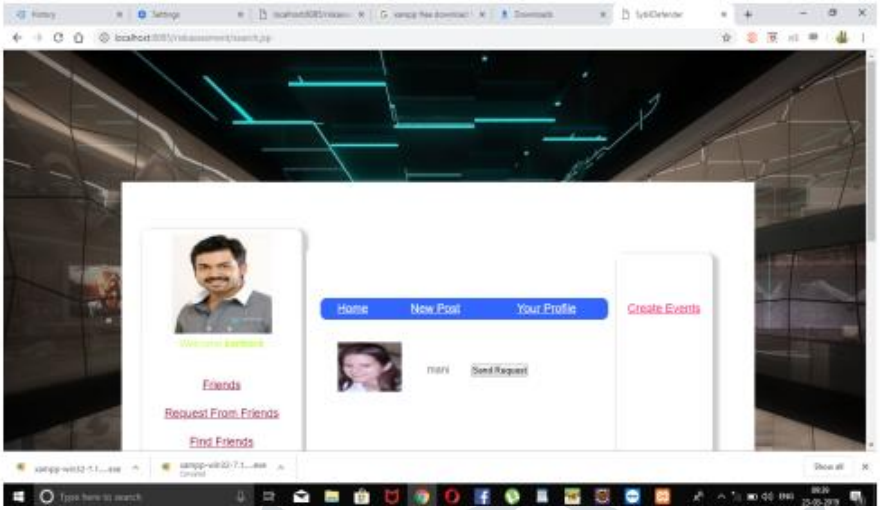
I proposed a two-phase risk assessment approach able to assign a risk score to each OSN user. This risk estimation is based on user's behavior under the idea that the more this diverges from what it can be considered as a 'normal behavior', the more the user should be considered risky. Experiments carried out on a real Facebook dataset show the effectiveness of our proposal. We plan to extend this work according to several directions. An interesting future work is the extension of the proposed two-phase risk assessment so as to make it able to perform a continuous monitoring and estimation of risk scores. Moreover, we plan to revise the risk assessment model so as to being deployable in Decentralized Online Social Networks, which are characterized by the absence of a central source of data to be analyzed. This will require to investigate decentralized data mining algorithms to gather user features.

OUTPUT:

LOGIN PAGE

PROFILE HOME

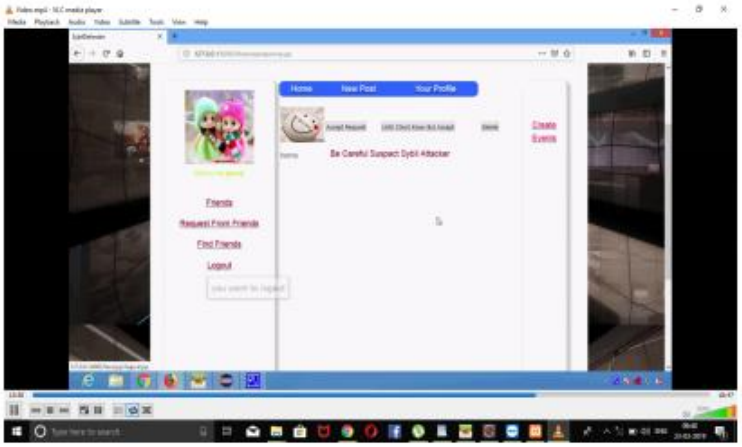
FINDING FRIENDS PAGE



ATTACKER DETECTION

REQUEST FROM FRIENDS

SYBIL ATTACKER SUSPECT



REFERENCES:

1] Cuneyt Gurcan Akcora, Barbara Carminati, and Elena Ferrari. Privacy in social networks how risky is your social graph? InData Engineering (ICDE), 2012 IEEE 28th International Conference

on, pages 9–19. IEEE, 2012.

[2] Christa SC Asterhan and Tammy Eisenmann. Online and faceto-face discussions in the classroom: A study on the experiencesof'active'and'silent'students. In Proceedings of the 9th internationalconference on Computer supported collaborative learning-Volume 1,pages 132–136. International Society of the Learning Sciences,2009.

[3] Leyla Bilge, Thorsten Strufe, Davide Balzarotti, and Engin Kirda.All your contacts are belong to us automated identity theft attackson social networks. In Proceedings of the 18th international conference on World wide web, pages 551–560. ACM, 2009.

[4] Yazan Boshmaf, Konstantin Beznosov, and Matei Ripeanu. Graphbased sybil detection in social and information systems. In Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on, pages 466–473. IEEE, 2013.

[5] Yazan Boshmaf, Dionysios Logothetis, Georgos Siganos, Jorge Ler´ıa, Jose Lorenzo, Matei Ripeanu, and Konstantin Beznosov. ´Integro: Leveraging victim prediction for robust fake account detection in osns. In Proc. of NDSS, 2015.

[6] Yazan Boshmaf, Ildar Muslukhov, Konstantin Beznosov, and Matei. [6] Yazan Boshmaf, Ildar Muslukhov, Konstantin Beznosov, and Matei Ripeanu. The socialbot network- when bots socialize for fame and money. In Proceedings of the 27th Annual Computer Security Applications Conference, pages 93–102. ACM, 2011.

[7] Paul S Bradley, Usama Fayyad, and Cory Reina. Scaling em (expectation-maximization) clustering to large databases. Technical report, Technical Report MSR-TR-98-35, Microsoft Research Redmond, 1998.

[8] Qiang Cao, Michael Sirivianos, Xiaowei Yang, and Tiago Pregueiro. Aiding the detection of fake accounts in large scale social online services. In NSDI, pages 197–210, 2012.

[9] George Danezis and Prateek Mittal. Sybilinfer- detecting Sybil nodes using social networks. In NDSS, 2009.

[10] Vacha Dave, Saikat Guha, and Yin Zhang. Measuring and fingerprinting click-spam in ad networks.