International Journal of Distributed and Cloud Computing
6 (1) June 2018 01-10
http://www.publishingindia.com/ijdcc

Reinforcing and Detecting Clone Profiles in Online Social Networks

D. Dave
Senior Faculty- IT Department, Ajeenkya D.Y. Patil University, Pune, Maharashtra, India.
Email: deeptidave09@gmail.com

Abstract: Online Social Networking (OSN) has brought real lives of people into the virtual world, lending itself to a wide variety of uses, but at the same time leaving its users vulnerable. OSN users make a great deal of personal information public in their profiles, often neglecting security precautions. An attacker takes advantage of the fact and creates clone profiles. When these clone profiles become active, the attacker sends friend requests to the victims' friends, inviting them to join the clone's network. If they do so, the attacker is in a position to cause annoyance and mischief to the genuine user. It is extremely difficult for other users to identify clone identities, as most of the attribute values are identical between genuine and clone profiles. This paper focuses on and seeks to contribute to the solution of the problem of profile clone attacks. It does so in two steps. First, it presents a potential form of clone attack called 'reinforced snowball sampling clone attack'. The experimental results of reinforced snowball sampling performed on Facebook showed that in-out friend networks, with the highest acceptance level for friend requests, are more susceptible to attack when request message is sent in native language. The paper goes on to present a novel framework of a clone recognizer called Clone Detector which depicts three effectual clone identification techniques. The Clone Detector uses a combination of content related and content free techniques to efficiently detect clone accounts on various social networks.

Keywords: Content free techniques, Content related, Online social network, Profile clone attack, Reinforced snowball sampling clone attack.

I. Introduction

With the growth of World Wide Web and new communication technologies, the market for online social networks has grown exponentially. Millions of people around the globe get connected with family, friends, and old school or college mates and are able to make new friends spread over huge geographical areas by just creating a profile, usually free of charge, on one or more of the several social networking websites. They are able to share information, photographs, thoughts and feelings with one another on a daily basis. The social networking sites accumulate users' personal data, monitor their activities, store their conversations and use them for marketing, sales, research, advertising and other purposes. The managers of social networking sites have a high responsibility to provide security to customers' profiles. Most of these websites have security mechanisms to safeguard client information against hackers, spammers, bots, malware, phishing attacks, profile clone attacks, cross site profile clone attacks and many more. However, since online users are frequently either unaware or unmindful of possible threats, attackers have it easy.

One of the most prominent type of attacks is profile clone attack, where the adversary steals user information and makes a new clone profile on the same social network (profile clone attack) or on some other social network (cross site profile clone attack) where the user is not registered. Another attack similar to clone attack is Sybil attack, where the attacker creates multiple identities (Sybil individuals) and tries to fetch all the important details of the online user. Lei Jin, Xuelian Long, Hassan Takabi and James B.D. Joshi [1] differentiated Sybil attacks from clone attacks on the basis of pre-requirements, network topology and attack impact. As Sybil and clone attacks seem alike, network administrators often confuse them and fall short of offering adequate security against them.

To detect multiple identities in a social network, Kahina Gani, Hakim Hacid and Ryan Skraba [2] used an unsupervised approach on authorship analysis and two clustering algorithms: *k-means* and *Kohonen Maps* to group data into clusters which had high similarity. The method consisted of three layers: *Representation Space, Learning layer* and *Validation.* In the results it was seen that, k-means was more permissive than Kohonen maps as it presented a prominent cluster containing most of the identities. Several other techniques have been proposed to detect compromised OSN accounts. Manuel Egele, Gianluca Stringhini, Christopher Kruegel and Giovanni Vigna [3] developed a mechanism to detect compromised accounts, which made use of both statistical models and anomaly detection, to monitor the movement of users on Twitter and Facebook. They developed a prototype model called *COMPA* to analyze a large stream of messages. Experimental results showed that COMPA detected compromised accounts with high precision.

Profile cloning is intensifying on OSN rapidly. Social network managers face the problem of identifying and neutralizing clone profiles in order to protect genuine users and, ultimately, the future of OSN. The problem can be simply stated as follows: when a new friend request comes to a user, he has no way of knowing whether it is a bona fide request from a genuine user or not. This paper presents a tool called *Clone Detector* which uses a combination of content related and content free techniques to identify clones inside the same social network in the quickest time possible.

The rest of the paper is organized as follows. Section 2 presents the basics of online social networking and clone attacks. Section 3 discusses the reinforced snowball sampling clone pattern techniques. Section 4 shows the experimental analysis of various forms of clone attack carried out on Facebook. Section 5 describes three hybrid clone detection techniques. Section 6 illustrates some related work performed by different researchers in the past few years. Finally, section 7 concludes the paper and includes a brief outline of future research work.

II. Online Social Networks And Clone Attacks

User profiles on an online social network display real-life personal data in the form of a virtual community. The offline associations of the user among his folks, groups and society can be clearly seen on the client's online account. Boyd and Ellison [4] defined online social networks as "web based services that allow individuals to (1) construct a public or semi-public profile within a bounded system, (2) articulate a list of other users with whom they share a connection and (3) view and traverse their list of connections and those made by others within the system". Various types of relationships exist in an OSN. Broadly, a user can have two types of relationships: *user-user link* and *user-group link*. Two friends are related to each other via user-user link and if a user has joined some community or group, he is connected with it through user-group link.

In a profile clone attack, the aim of the adversary is to fetch the details of the victim through sniffing and use them for creating a new profile on the same OSN. The attacker presents the new profile as a valid one and dupes the victim's friends into accepting friend requests from the cloned profile. Having succeeded in doing so, he proceeds to post malicious texts and images in the victim's name.

III. Reinforced Snowfall Profile Clone Attack

In the traditional attack pattern, the attacker would simply create some random profiles and send a number of friend requests arbitrarily. These fake accounts were easily detected and blocked by other users, as they did not contain mutual friends or common attributes such as family members, school, working place, city, country, etc. The phenomenon of cloning attack came later and proved much more intractable. It took two forms: clone identities were built within the same OSN structure (profile clone attack) or on some other OSN where the user was not registered (cross-site profile clone attack).

Recently in 2013, Zifei Shan, Haowen Cao, Jason Lv, Cong Yan and Annie Liu [5] conceived of a new enhanced version of cloning known as s*nowball sampling* [13] and *iteration attack*. The weaknesses of the original clone attacking pattern were eliminated in these two types of techniques. These methods use the combination of clone attack and traditional attack pattern which is generally adopted by spammers. Snowball sampling is an iterative technique which completely utilizes its friends' network, which consists of the victim's friends who have accepted the friend request from the cloned profile. When some friend of the victim acknowledges the request from the cloned profile, the attacker sends friend request to the friends of this person. As the number of mutual friend's increases, the authenticity of this clone profile grows and more people accept the friend requests. In the case of iteration attack, the adversary gains access to users information and then introduces multiple Sybil profiles in the OSN. In this way, the adversary significantly influences the network and uses these Sybil accounts for malicious purpose such as phishing, spreading viruses, advertisements, backdoors, spamming and many other harmful actions.

The primary modus operandi of the adversary in a clone attack is to send friend requests, his objective being to build up a relationship of trust with a number of people. This paper introduces a novel attack pattern to be called *reinforced snowball sampling profile clone attack* where features of snowball sampling are combined with friend network systems. Generally, each user has four types of friend lists: regularly added friends' list, in-out friends' list, recommended friends' list and excluded friends' list. The *added regular friend list* constitutes of those, who are already present in the user's profile and are in regular contact. This might include family members, high school or college friends, work place colleagues and so on. Every user has some friends who are included in the network circle but with whom the user does not interact on a regular basis. These friends are called *in-out friends*. They are already inside the network but are treated as outsiders. The*re commended friendlist* contains names which the OSN site generates on the basis of common friends, similar interests, identical educational backgrounds or common work experience. The *excluded friendlist* encloses those persons whom the user does not want to add to his network for some reason; e.g. to avoid social embarrassment. This list could include the user's family members, co-workers or company heads. Reinforced snowball sampling integrates the concept of multiple friend list types with snowball sampling.

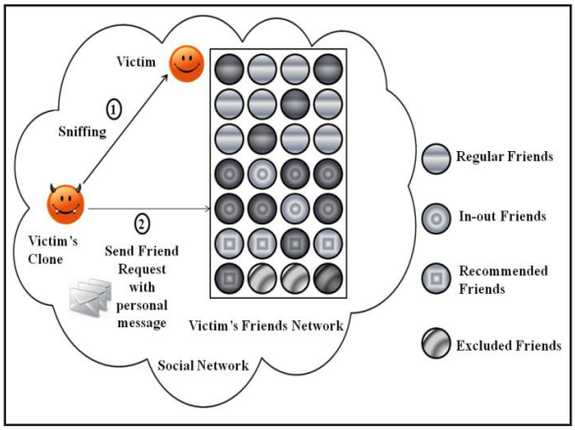

Fig. 1: Reinforced Snowball Sampling Profile Clone Attack

Fig. 1 depicts reinforced snowball sampling profile clone attack where the attacker creates a clone profile of the target by sniffing his or her profile, analyzes the target's friend network and then sends friend requests intelligently. As regular friends usually meet every day, there is high probability of alerting the victim; so the attacker sends only a few requests to them. In-out folks do not interact with the target daily but the attacker can still get quite good information about the victim with the least risk; therefore, he sends a fair amount of friend requests to in-out friends. Recommended friends and excluded friends are not inside the target's circle but it is assumed that in the near future they might get into contact with oneanother. Therefore, the attacker sends friend requests to them also, but choosing wisely. Invitations to join the clone's network are sent along with a personal message written in the native language of the victim. In Fig. 1, some of them (highlighted dark ones) accept the clone's request in the belief that it is sent from a second legitimate account of the genuine user.

IV. Experimental Analysis on Facebook

In order to validate the affectivity of reinforced snowball sampling profile clone attack, an experiment was conducted on the most popular social networking site, Facebook, with current membership of over a billion. Reinforced profile clone attack was compared with traditional attacks, original cloning and snowball sampling. 21 cloned accounts were manually created on Facebook for a week. Due to limitations of APIs, and existing security and complexities of Facebook, a much larger number of profiles could not be created. All the cloned profiles were created with the users' consent and were deactivated after the completion of the experiment.

A. Assessment

Zifei Shan et al [5] conducted their experiment on Renren, a Chinese social networking site and concluded that snowball sampling method can be a more dangerous tool for profile clone attacks. The present paper adds another parameter to snowball sampling, viz. friend network lists. This reinforced snowball sampling profile clone attack method was utilized in our experiment. Facebook user data consisting of username, birth date, email id, profile picture, hometown, current city, language known, and work and education information were used.

The reinforced snowball sampling profile clone attack was compared with traditional attack, original cloning attack and snowball sampling. In traditional attack, the adversary randomly sent friend requests without any personal message. This method was followed. In the original cloning pattern three different levels of profiles were identified on the basis of profile attribute similarity between the original victim's account and the cloned account:

- *Level 1* has the same name as the victim,
- *Level 2* shares the name, birth date, school information and language, and
- *Level 3* uses the same profile name, date of birth, school, city, language and photograph.

Three profiles were created under traditional attack pattern and friend requests were sent randomly to anybody in the network with no personal message. In original cloning attack, three profiles were created of each level, thus giving a total of nine cloned profiles in this section. In snowball sampling and reinforced snowball sampling profile clone attack, each of three cloned accounts were created but friend requests were sent by applying friend list networks. In snowball sampling (level 1), a text message in English language was sent along with friend requests for which 4 sets were created.

- *Set A* constituted of *regular friends* with whom the user is in daily or frequent contact.
- *Set B* included *in-out friends* i.e., those friends who were added into the victim's friend list but did not communicate frequently.
- *Set C* was made up of *recommend friends*. These requests were either recommended by OSN or were present in the incoming friend request section or recently included into the targets circle.
- *Set D* was composed of *excluded friend*s, who were kept out of the victim's network.

A similar format was established in reinforced snowball sampling cloning pattern (level 1) but the text was sent in the German language which the victim had never used in past. To further test the feasibility of reinforced profile clone attack with language being taken into consideration, its level was increased to 3 and the personal message was sent in the mother tongue of the victim. The message in English was: "Hi, I have created a new account. Please accept my friend request". The same text was sent in German and respective native languages of users.

All the users whose cloned profiles were created were active on Facebook and contained an average of 350 friends within their network. From each cloned profile, twelve friend requests were

sent. In traditional attack, a total of forty two friend requests were sent randomly from three accounts. In original cloning attack, forty two friend requests were sent from each level; while in snowball sampling and reinforced snowball sampling profile clone attack, twelve requests were sent according to Set A and Set B, and nine from Set C and Set D. After creating clone accounts for snowball sampling and reinforced snowball sampling, friend requests were sent to the victim's friends who accepted clones requests. This process went in an iterative fashion until all forty two requests were sent successfully.

V. Experimental Results and Analysis

The experiment on Facebook lasted for a week and subsequently analysis was performed. Table 1 represents various attack patterns of clone attack. Seven parameters were taken into consideration while conducting the experiment. Recommendation of friends is one of the significant features of Facebook and is provided by OSN on the basis of users' likes, number of mutual friends, joined communities and many other factors. Hash sign (#) implies the approximate value of number of recommendations to cloned profiles. Incoming requests symbolizes the number of friend requests sent to cloned profiles from persons whom we didn't sent requests. Number of friends alert indicates the alertness of friends while accepting invitations. Number of blocking designates the attentiveness of users who blocked the clone accounts in time. If a person is blocked on Facebook, then he can no longer see timeline posts of the user who blocked him. Also, he cannot tag on any photograph, chat, send invitations to join group, communities, or he cannot even send a friend request to the user who blocked him. Asterisk (*) shows that, when clones invitation was sent to victims friends, they spotted the cloned profile and immediately notified it to the target user.

The first column of Table I shows traditional sybil attack, where only an average of 0.095 requests were accepted and due to low availability of users information, very few (750 approx.) recommendations came. In the original cloning attack, as the profile visibility increased at the 3rd level, the acceptance rate also increased. Here the number of recommendations also increased because more information was available with the OSN provider. In initial levels, a number of friends alert and profile blocking were also observed. In snowball sampling, where invitations were sent along with a personal message in English, acceptance rate improved further along with the number of recommendations. On the other hand, while considering reinforced snowball sampling (text in German language), acceptance rate and number of recommendations were reduced and friends alert and blocking of accounts increased when the friend list network was applied to reinforce clone attack in level 3, there was a sudden rise in acceptance. It can be noticed that in Set B where all in-out friends of victim were sent invitations, more than 90% users acknowledged the requests without any alerts and blocking. Hence, it can be concluded from the results that reinforced snowball sampling profile clone attack when applied on in-out friends can be more harmful than any other clone attack. It can be further inferred that the combination of snowball sampling and in-out friend list network, when used along with a message in the native language, enhances the friend acceptance rate, with least possibility of the target being alerted by those getting friend requests.

Table I: Experimental Analysis of Various Clone Attack Patterns

Attack Pattern → / Parameter ↓	Traditional cloning (no message)	Original cloning attack			Snowball sampling Level-1 (message in English)				Reinforce Snowball sampling Level-1 (message in German)				Reinforce Snowball sampling Level-3 (message in native language)			
		Level 1	Level 2	Level 3	Set A	Set B	Set C	Set D	Set A	Set B	Set C	Set D	Set A	Set B	Set C	Set D
No. of friend requests sent	42	42	42	42	12	12	9	9	12	12	9	9	12	12	9	9
Accepted Requests	4	14	19	29	8*	9	5	4	3*	5*	1*	0	10	11	7	5
Average of Accepted Requests	0.095	0.333	0.452	0.690	0.667	0.750	0.556	0.444	0.250	0.417	0.111	0	0.833	0.917	0.778	0.556
Recommendati-ons from Facebook	750#	800#	990#	1005#	1050#				500#				2150#			
Incoming Requests	0	0	1	6	4				0				10			
No. of friends alert	0	6	4	2	4				5				1			
No. of blocking	0	1	1	0	0				1				0			

* Clone profile identified by the victim's friends and the victim informed about his cloned profile, # approximate value

VI. Clone Detector

Clone accounts can be identified using either content-based or content-free approaches. Early authors used to take advantage of user generated information to detect cloned profiles. Content based process could be implemented by applying attribute similarity, friend network similarity and profile analysis over

a period of time. These techniques required low authority and were quite easily implemented. But, by the time clones are detected within an OSN, the adversary would have already gained essential information about the target. To overcome the weaknesses of content related methods, recently in 2013 authors [5] proposed a light weighted, real time detector called CloneSpotter which utilized recent IP (Internet Protocol) list of the user to detect the clones. This paper presents a new clone recognizer called Clone Detector which uses three hybrid dynamic approaches and takes advantage of both content base and content free mechanisms. Different threshold conditions are applied on various parameters, but initially attribute values are matched as the adversary sniffs the victim's profile and tries to make the clone account as authentic as possible. If a particular user's attribute values pass the threshold parameters ($\lambda = 5$) then further conditions are applied and results are obtained. The following subsections describe three clone detection algorithms.

A. Attribute Similarity, IP Sequence and Login Time Pattern

Most of the OSN servers, store user generated information in the form of cookies, users' IP addresses and login time are a few of them. Authors [5] exploited only four recent IP addresses of the user and on the basis of that, detected the clone. Here another parameter has been added, namely, *login time pattern,* assuming that each user has the facility to use only a single browser and can login with one profile at a time. It is also supposed that a user having two accounts on the same OSN will have same IP sequence addresses. The clone identifying algorithm of Clone Detector is explained in Algorithm 1.

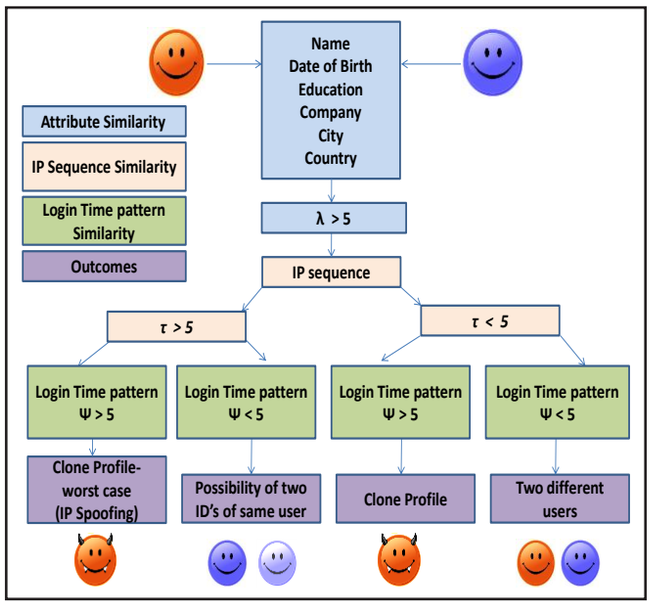

Fig. 2: Detecting Clones through Attribute Similarity, IP Sequence and Login Time Pattern

ALGORITHM 1: Attribute similarity, IP and Login Time Pattern

Input:-
U: The set of all users.
$Friend_x$: Friend list of user x ($x \in U$).
$Profile_x$: Profile information of user x ($x \in U$).
$Similarity_{x,y}$: Similarity of profile x and y ($x, y \in Profile$).
ASM_{xy}: Attribute string matching of user x and y ($x, y \in Profile$).
Attribute Threshold (λ) = 5
Education = 2pts, Date of Birth=2pts, Company=2pts, City=1pt and Country=1pt.
$RIPS_x$: Recent Login IP sequence of user x ($x \in U$).
Recent login IP sequence threshold (τ) = 5.
LTP_x: Login Time Pattern action of user x ($x \in U$).
Login Time Pattern threshold (ψ) = 5.

Procedure:-
for all friend requests from A to B (A, B $\in$ U)
do
  for u $\in$ $Friend_B$ do,
    If u.name = A.name then
for $Similarity_{profile}$, $Similarity_{profile}$
If > $\{ASM_{profileu} = ASM_{profileA}\} > \lambda$
    AND $\{\cap\}$ $\{RIPS_u \cap RIPS_A\} > \tau$
    AND $\{LTP_u \cap LTP_A\} > \psi$ then
    Return A is a clone
    endif
If > $\{ASM_{profileu} = ASM_{profileA}\}$ λ
    AND $\{RIPS_u \cap RIPS_A\} > \tau$
    AND $\{LTP_u \cap LTP_A\} < \psi$ then
    Return two ID's of same user
    endif
If $\{ASM_{profile} = ASM_{profileA}\} > \lambda$
    AND $\{RIPS_u \cap RIPS_A\} < \tau$
    AND $\{LTP_u \cap LTP_A\} > \psi$ then
    Return A is a clone with no IP spoofing
    endif
    If $\{ASM_{profileu} = ASM_{profileA}\} > \lambda$
    AND $\{RIPS_u \cap RIPS_A\} < \tau$
    AND $\{LTP_u \cap LTP_A\} < \psi$ then
    Return two different users
    endif
  endfor
 endif
endfor
endfor

When user *A* sends friend request to user *B,* user *B* checks if there is any other user (user u) already added with the name *A*. If same named user is already in the friend list, then profile similarity check is performed between user *A* and *u*, by using attribute string matching. For each attribute match, points

are assigned. For education, Date of Birth, Company name, City and Country; following points are allocated 2 points, 2 points, 2 points, 1 point and 1 point respectively. Sum of attribute similarity must be at least 5 for further clone detection mechanisms. Sometimes, string matching fails as the way of writing differs. For example, United States of America can also be written as USA. Here string matching does not match these words and hence it does not succeed. To overcome this problem, comparison of strings on Google search engine has been applied. If both the attribute values appear in top three hits of the Google search list, then both of the strings are declared equivalent. Along with attribute string, IP address and Login time pattern are matched. Here, four cases could arise: when $\tau > 5$ and $\psi > 5$, then it is considered as a worst case (least probability) in which attacker spoofs the IP address and correlates timing pattern also. When $\tau > 5$ but $\psi < 5$, then it shows that same user has two different accounts on the same OSN. While if $\tau < 5$ and $\psi > 5$ situation arises, then it has the highest probability of Clone attack because adversary can easily spoof the login timings but it's difficult to match the IP addresses. Finally, when $\tau < 5$ and $\psi < 5$, then it is declared that these two profiles belong to two separate users.

B. Attribute Similarity, IP Sequence and Friend Network Similarity

Every user has a specific type of companion circle on OSN. LinkedIn is generally used for business purposes and professionals are added, while Flickr is used for photo sharing use, where personal networks can be established. For browsing any social network, each user is allotted an IP address. If a threshold is put on sending invitations on each IP address, then clones can be quickly and easily be detected. Fig. 3 and algorithm 2 described below, explains this phenomenon.

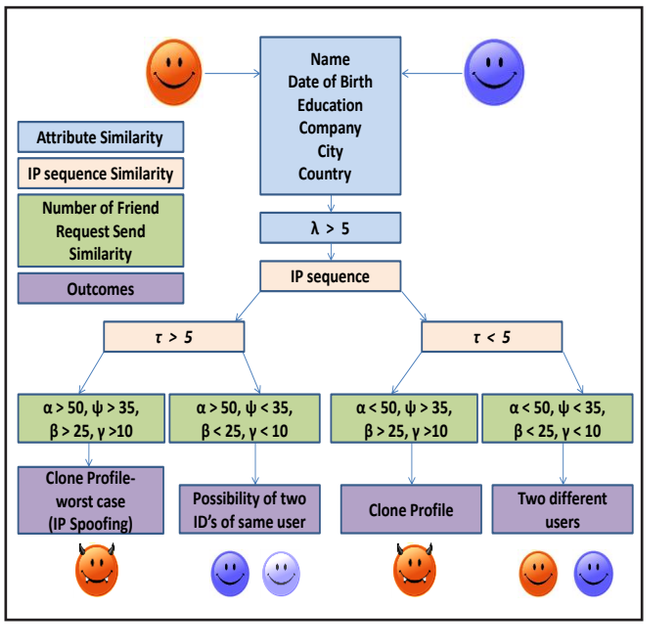

Fig. 3: Detecting Clones through Attribute Similarity, IP Sequence and Friend Network Similarity

ALGORITHM 2: Attribute similarity, IP and Friend Network Similarity

Input:-

U: The set of all users.

$Friend_x$: The friend list of user x ($x \in U$).

$RIPS_x$: Recent Login IP sequence of user x ($x \in U$).

$Profile_x$: Profile information of user x ($x \in U$).

$Similarity_{x,y}$: Similarity of profile x and y ($x, y \in Profile$).

ASM_x: Attribute string matching of user x ($x \in U$).

Attribute Threshold (λ) = 5

Education= 2pts, Date of Birth=2pts, Company=2pts, City=1pt and Country=1pt.

FR_x: Number of friend request sent by user x to regular friends of victim ($x \in U$).

Friend Threshold (α) = 50

$IOFR_x$: Number of friend requests sent by user x to in-out friends of victim ($x \in U$).

In-out friend threshold (φ) = 35

RFR_x: Number of friend request sent by user x to recommended friends of victim ($x \in U$).

Recommend friend threshold (β) = 25

EFR_x: Number of friend request sent by user x to excluded friends of victim ($x \in U$).

Excluded friend threshold (γ) = 10

Procedure:-

for all friend requests from A to B (A,B $\in$ U)

do

for u $\in$ $Friend_B$ do,

If u.name = A.name then

for $Similarity_{profileu}$, $Similarity_{profileA}$

If $\{ASM_{profileu} = ASM_{profileA}\} > \lambda$

AND $\{RIPS_u \cap RIPS_A\} > \tau$

AND$\{ FR_u = FR_A > \alpha\} \cap \{ IOFR_u = IOFR_A) > \varphi\} \cap \{RFR_u = RFR_A) > \beta\} \cap \{EFR_u = EFR_A > \gamma\}$ then

Return A is a clone

endif

If $\{ASMprofile_u = ASM_{profileA}\} > \lambda$

AND$\{RIPS_u \cap RIPS_A\} > \tau$

AND$\{FR_u = F_{RA} > \alpha\} \cap \{IOFR_u = IOFR_A < \varphi\} \cap \{(RFRu = RFR_A < \beta\} \cap \{EFR_u = EFR_A < \gamma\}$ then

Return two ID's of same user

endif

If$\{ASM_{profileu} = ASM_{profileA}\} > \lambda$

AND$\{RIPS_u \cap RIPS_A\} < \tau$

AND$\{FR_u = FR_A < \alpha\} \cap \{(IOFR_u = IOFR_A) > \varphi\} \cap \{RFR_u = RFR_A > \beta\} \cap \{EFR_u = EFR_A > \gamma\}$ then

Return A is a clone

endif

If$\{ASM_{profileu} = \text{ASM}_{profileA}\} > \lambda$

AND $\{RIPS_u \cap \underline{RIPS_A}\} < \tau$

AND$\{FR_u = FR_A < \alpha\} \cap \{IOFR_u = IOFR_A < \varphi\} \cap \{(RFR_u = RFR_A < \beta\} \cap \{EFR_u = EFR_A < \gamma\}$ then

Return two different users

endif

endfor

endif

endfor

endfor

Clone Spotter [5] identified clones by using recent list of IP addresses. Here another constraint of friend network similarity is added to it. When an invitation comes from user *A* to user *B*, it is initially checked if there is another user (user u) with the same name. Similarity between users' attributes is matched and points are assigned to them accordingly. Further on, recently used IP address and friend network similarity thresholds are placed. If $\lambda > 5$, $\tau > 5$, $\alpha > 50$, >35, $\beta > 25$ and $\gamma > 10$ then it is considered as a worst case (least probability) of clone attack. If $\lambda > 5$, $\tau > 5$, $\alpha > 50$, < 35, $\beta < 25$ and $\gamma < 10$ then in this case it is believed that same user has two accounts because same user will not send friend requests to more than 10 excluded friends. If $\lambda > 5$, $\tau < 5$, $\alpha < 50$, > 35, $\beta > 25$ and $\gamma > 10$ is the condition, then it has the highest possibility of clone attack because intelligent adversary generally will not send more than 50 invitations to close folks of victim as they are in regular contact with him, so there is a good chance of quick intimation of clone attack to user. When $\lambda > 5$, $\tau < 5$, $\alpha < 50$, < 35, $\beta < 25$ and $\gamma < 10$ event occurs, then it is affirmed that there exist two different genuine users. The drawback with this approach is that, if the adversary gets to know all the threshold values then it would become difficult to detect clones.

Attribute Similarity, IP Sequence and Trusted Relationships on Friend Network Similarity

To make Clone Detector more effective and efficient to spot clones on OSN, this research work proposes a novel framework which validates a user only if he or she gives correct answers to some queries on each friend request. If a genuine user creates a second account on the same OSN and sends invitations to the same old friends, he/she should be able to answer some personal questions with the same IP address. Fig. 4 and algorithm 3 describes the following technique.

ALGORITHM 3: Attribute similarity, IP sequence and trusted relationships on friend network similarity

Input:-

U: The set of all users.

$Friend_x$: The friend list of user x ($x \in U$).

$RIPS_x$: Recent Login IP sequence of user x ($x \in U$).

$Profile_x$: Profile information of user x ($x \in U$).

$Similarity_{x,y}$: Similarity of profile x and y ($x,y \in Profile$).

ASM_x: Attribute string matching of user x ($x \in U$).

Attribute Threshold (λ) = 5

Education=2pts, Date of Birth=2pts, Company=2pts, City=1pt and Country=1pt.

FR_x: Number of friend request sent by user x to regular friends of victim ($x \in U$).

Friend Threshold (α) = 50

TFR_x: Trusted friend relationship of user x ($x \in U$).

Security question on Friend threshold (η) = 15

$IOFR_x$: Number of friend requests sent by user x to in-out friends of victim ($x \in U$).

In-out friend threshold (φ) = 35

$TIOR_x$: Trusted In-out relationship of user x ($x \in U$).

Security question on in-out friend threshold (ζ) = 12

RFR_x: Number of friend request sent by user x to recommended friends of victim ($x \in U$).

Recommend friend threshold (β) = 25

TRR_x: Trusted Recommend relationship of user x ($x \in U$).

Security question on Recommend friend threshold (ρ) =10

EFR_x: Number of friend request sent by user x to excluded friends of victim ($x \in U$).

Excluded friend threshold (γ) = 10

TER_x: Trusted excluded relationship of user x ($x \in U$).

Security question on Excluded friend threshold (ω) = 5

Procedure:-

for all friend requests from A to B ($A,B \in U$)

do

for u $\in$ *$Friend_B$* do,

If u.name = A.name then

for *$Similarity_{profileu}$*, *$Similarity_{profileA}$*

If$\{ASM_{profileu} = \text{ASM}_{profileA}\} > \lambda$

AND$\{RIPS_u \cap RIPS_A\} > \tau$

AND$\{FRu = FR_A > \alpha\} \cap \{IOFR_u = IOFR_A > \varphi\} \cap \{\text{RFR}_u = RFR_A) > \beta\} \cap \{(EFR_u = EFR_A < \gamma\}$

AND$\{TFR_u = TFR_A > \eta)\} \cap \{TIOR_u = TIOR_A > \zeta\} \cap \{TRR_u = TRR_A > \rho\} \cap \{TER_u = TER_A > \omega\}$ then

Return genuine user having two accounts

endif

If $\{ASM_{profileu} = ASM_{profileA}\} > \lambda$

AND $\{RIPS_u \cap RIPS_A\} > \tau$

AND $\{FR_u = FR_A < \alpha\} \cap \{IOFR_u = IOFR_A > \varphi\} \cap \{RFR_u = RFR_A > \beta\} \cap \{EFR_u = EFR_A > \gamma\}$

AND $\{TFR_u = TFR_A > \eta\} \cap \{TIOR_u = TIOR_A > \zeta\} \cap \{TRR_u = TRR_A < \rho\} \cap \{TER_u = TER_A < \omega\}$ then

Return A is clone

endif

If $\{ASM_{profileu} = ASM_{profileA}\} > \lambda$

AND $\{RIPS_u \cap RIPS_A\} < \tau$

AND $\{FR_u = FR_A > \alpha\} \cap \{IOFR_u = IOFR_A > \varphi\} \cap \{RFR_u = RFR_A > \beta\} \cap \{EFR_u = EFR_A < \gamma\}$

AND $\{TFR_u = TFR_A > \eta\} \cap \{TIOR_u = TIOR_A > \zeta\} \cap \{TRR_u = TRR_A > \rho\} \cap \{TER_u = TER_A > \omega\}$ then

Return same user login at different time interval

endif

If $\{ASM_{profileu} = ASM_{profileA}\} > \lambda$

AND $\{RIPS_u \cap RIPS_A\} < \tau$

AND $\{FR_u = FR_A < \alpha\} \cap \{IOFR_u = IOFR_A < \varphi\} \cap \{RFR_u = RFR_A < \beta\} \cap \{EFR_u = EFR_A < \gamma\}$

AND $\{TFR_u = TFR_A < \eta\} \cap \{TIOR_u = TIOR_A < \zeta\} \cap \{TRR_u = TRR_A < \rho\} \cap \{TER_u = TER_A < \omega\}$ then

Return two different users

endif

endfor

endif

endfor

endfor

With the help of trusted relationships limitation, performance of Clone Detector becomes more cost effective. Though asking of personal questions becomes difficult, but it can prove to be highly effectual and can yield better clone detection. Here also four cases evolve. In the first case, when $\alpha > 50$, $\varphi > 35$, $\beta > 25$, $\gamma < 10$, $\eta > 15$, $\zeta > 12$, $\rho > 10$ and $\omega > 5$ there is likelihood that genuine user has two profiles because genuine user will not send more than 10 requests to excluded friends as they usually avoid adding some person into their network. When $\alpha > 50$, $\varphi < 35$, $\beta < 25$, $\gamma < 10$, $\eta > 15$, $\zeta > 12$, $\rho < 10$ and $\omega < 5$ happens, there is high risk of profile cloning attack. This is so because the adversary might be able to answer security questions related to close friends of the victim but it becomes difficult for attacker to answer questions related to recommend and excluded friends as they are not much in contact with target. If τ becomes less than 5 and $\alpha > 50$, $\varphi > 35$, $\beta > 25$, $\gamma < 10$, $\eta > 15$, $\zeta > 12$, $\rho > 10$ and $\omega > 5$ then, there are chances that the same user would have logged in at different time intervals. In the last case, when $\alpha < 50$, $\varphi < 35$, $\beta < 25$, $\gamma < 10$, $\eta < 15$, $\zeta < 12$, $\rho < 10$ and $\omega < 5$ are present then it is stated that two profiles belongs to two distinct users.

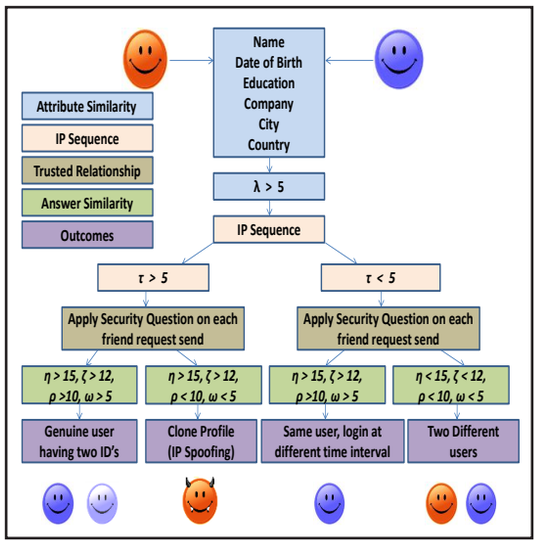

Fig. 4: Detecting Clones through Attribute Similarity, IP Sequence and Trusted Relationships on Friend Network Similarity Measure

D. Merits and Demerits of Clone Detector

This section discusses the evaluation of Clone Detector, its advantages and its drawbacks. All users' information is stored on OSN servers and the parameters which have been taken into consideration are stored on servers in the form of cookies. Clients attribute values, login time, IP addresses, friend lists, communication messages; URL's exploited, etc. are some of them. Here Clone Detector has considered the most probable scenarios that attackers can exploit. Some other situations could also arise with different threshold values, but they would have the least probability; so we have ignored them. The main benefit of implementing Clone Detector is real time detection. All the algorithms described above have made use of content free information. Since these characteristics are not dependenton user generated information,clone identification can be achieved in comparatively little time.

Sometimes, string matches fail as the manner of writing differs from person to person. To overcome this weakness, Google search mechanism has been applied which compares top three hits of both strings and takes decisions. This makes our Clone Detector more efficient to use. Another asset of Clone Detector is its cost. A very low overhead is obtained on space and time complexity. As the elements used for detection are quite small in number and yet efficient in identifying clones, so the overall complexity of Clone detector comes out to be quite low.

The drawbacks with Clone Detector are that, sometimes it might suffer from IP spoofing but that will be considered as a

worst case where the attacker will have to be a highly qualified professional. Another downside of Clone Detector is that, if the adversary gets to know all the threshold values which this research paper has used in three algorithms, he might try to deceive our Clone Detector.

VII. Related Work

Leyla Bilge, Thorsten Strufe, Davide Balzarotti and Engin Kirda [7] discovered a new threatening technique of automated identity theft on five social networking sites (Facebook, XING, StudiVZ, MeinVZ and LinkedIn). They implemented their architecture on a prototype system called iCloner (identity cloner) where CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) breaking mechanism was also built up. They also provided the scoring system for detecting cross site profile clone attack which has been utilized in our Clone Detector scheme.

In 2011, Georgios Kontaxis, Iasonas Polakis, Sotiris Ioannidis and Evangelos P. Markatos [8] presented a procedure of identifying clones on LinkedIn social network, based on users' attribute value similarity. This feature of matching attributes of genuine user and suspected user has been exercised in our process also. They cloned 10 existing LinkedIn accounts of their laboratory members and were able to detect all the clone profiles without any false positive and false negative. In addition, authors [9] used friend network similarity (actual, recommend and excluded friend) concept to find out clones on Facebook. This friend network similarity concept has been modified here by adding another type of friend list called as in-out friend network. The shortcomings of this paper was that they used offline dataset of Facebook and their approach failed to identify clone profiles of those clients who never used Facebook for social networking. Mauro Conti, Radha Poovendran and Marco Secchiero made a study to analyze the growth rate graphs of social networks from a dynamic point of view. They considered three parameters: number of OSN friends evolved over a period of time, real social communication and structure of OSN graph evolved over a time period. The weakness with their approach was they were not able to fully reconstruct day by day graph of the social network from the day of account creation. Various other methods [11] [12] of detecting compromised accounts have been proposed. Authors [11] utilized tweet timestamps value while researchers [12] combined statistical model with anomaly detection.

VIII. Conclusion And Future Work

Security of client data on Online Social Networks has become the greatest concern of social network providers. This paper presents an enhanced version of snowball sampling namedreinforced snowball sampling profile clone attack. The results indicate that, if more attributes of the user are made public (level 3) and if message communication is performed in the native language, then there is a high probability of acceptance of friend requests (11 friend requests accepted for Set B) from in-out friend networks. In addition, this paper manifests three new dynamic techniques of detecting clone profiles by implementing Clone Detector. These techniques employ a hybrid method, which takes advantage of both content related and content free processes. The property of real time detection makes Clone Detector a more powerful tool which can be easily used by OSN providers.

In future, Clone Detector techniques can be practically implemented on any social networking website. Some additional parameters can be included for identifying clones. For example, Click pattern of genuine user, languages and URLs used by the normal user and the suspicious user can be analyzed by observing the activities performed by them. OSN server collects plenty of cookies of individual users. Cookies storing recent pages surfed, current location, operating system employed and browser used can also be exploited to detect clones on Online Social Networks.

References

[1] L. Jin, X. Long, H. Takabi, and J. B. D. Joshi, "Sybil attacks vs. identity clone attacks in online social networks," *In Proceedings 6th International Conference on Information Security and Assurance*, Shanghai, China, pp. 28-30, April 2012.

[2] K. Gani, H. Hacid, and R. Skraba, "Towards multiple identity detection in social networks," *In Proceedings of the 21st International Conference Companion on World Wide Web, USA,* pp. 503-504, 2012.

[3] M. Egele, G. Stringhini, C. Kruegel, and G. Vigna, "COMPA: Detecting compromise accounts on social networks," *In Proceedings of the 2013 ISOC Network and Distributed System Security Symposium*, 2013.

[4] D. M. Boyd, and N. B. Ellison, "Social network sites: Definition, history, and scholarship," *Journal of Computer- Mediated Communication,* vol. 13, no. 1, pp. 210-230, 2007.

[5] Z. Shan, H. Cao, J. Lv, C. Yan, and A. Liu, "Enhancing and identifying cloning attacks in online social networks," *In Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication, Article No. 59, ACM, Kota Kinabalu, Malaysia*, pp.17-19, 2013.

[6] L. Jin, H. Takabi, and J. B. D. Joshi, "Towards active detection of identity clone attacks on online social networks," *In Proceedings of the 1st ACM Conference on Data and Application Security and Privacy,* USA, pp. 27-38, February, 2011.

[7] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda, "All your contacts are belong to us: Automated identity theft attacks on social networks," *In Proceedings of the 18th International Conference on World Wide Web ACM, New York, USA*, pp. 551-560, 2009.

[8] G. Kontaxis, I. Polakis, S. Ioannidis, and E. P. Markatos, "Detecting social network profile cloning," *In Proceedings of the 3rd International Workshop on Security and Social Networking,* pp. 295-300, 2011.

[9] L. Jin, H. Takabi, and J. B. D. Joshi, "Towards active detection of identity clone attacks on online social networks," *In Proceedings of the 1st ACM Conference on Data and Application Security and Privacy, USA,* pp.27-38, February 2011.

[10] M. Conti, R. Poovendran, and M. Secchiero, "FakeBook: Detecting fake profiles in on-line social networks," *In Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining,* pp. 1071-1078, 2012.

[11] C. M. Zhang, and V. Paxson, "Detecting and analyzing automated activity on twitter," *In Proceedings of the 12th International Conference on Passive and Active Measurement Springer-Verlag Berlin Heidelberg,* pp. 102-111, 2011.

[12] M. Egele, G. Stringhini, C. Kruegel, and G. Vigna, "COMPA: Detecting compromised accounts on social networks," *Proceedings of the 2013 ISOC Network and Distributed Systems Symposium, San Diego, CA*, pp. 24-27 February 2013.