VoIP Profiler: Profiling Voice Over IP User Communication Behavior

Sainath Batthalla, Mayank Swarnkar, Neminath Hubballi
Discipline of CSE, School of Engineering
Indian Institute of Technology Indore
{cse1200106, phd1401101001, neminath}@iiti.ac.in

Maitreya Natu
Tata Research Development and Design Centre
Pune, India
maitreya.natu@tcs.com

***Abstract*—Understanding the user behavior in Voice over IP (VoIP) communication has twofold advantages. It helps in detecting anomalies and also helps in planning VoIP infrastructure deployment and optimization. Anomalies arise out of various attacks and misuses like flooding, malformed messages and spam messages. In this paper we propose VoIP Profiler a method for profiling the VoIP activities at user level. For profiling users we identify a set of parameters and compute statistics of these parameters for each user using VoIP traffic. Subsequently we use these parameters to classify users (and detect anomalies). We simulate an enterprise network and experiment with a large scale VoIP dataset and identify different types of users with high success rate.**

Keywords*—Session Initiation Protocol, User Profiling, Anomaly Detection

I. Introduction

Internet telephony or Voice over IP (VoIP) is a cheaper alternative for telephone communication compared to traditional Public Switched Telephone Networks (PSTN). In addition to supporting voice calls VoIP also provides services like messaging, video calling, etc. Typically VoIP communication happens in two phases as signaling and data transmission. Signaling is used to establish and disconnect calls and in data transmission voice data is carried in the form of IP packets. There are various signaling protocols like H.323 and SIP [2] and others for this purpose. Of late, Session Initiation Protocol (SIP) has become a de-facto standard [5] signaling protocol because of its simplicity and easy implementation.

Session Initiation Protocol is an application layer signaling protocol which can establish, modify and terminate connection in Internet Telephony [1]. SIP being a text based protocol and with no in-built security mechanism is vulnerable to a number of attacks [14] like Denial of Service, User enumeration, Billing attacks, Connection hijacking, Spamming, etc. Denial of Service can be created either through flooding [7], [25] or by sending malformed messages [6]. There are works in the literature to detect these types of attacks. All of these methods require monitoring and analyzing SIP traffic. Hence a framework for collecting and analyzing SIP traffic is required. There have been attempts to understand the SIP and VoIP traffic at the network level by collecting traffic from open source VoIP servers [24], [13].

In this paper, we argue that in addition to knowing traffic behavior at network level, it is also important for a network administrator to understand the user behavior. Knowing type of users and their calling patterns will help in many ways. For example, knowing what percentage of calls are long distance calls, who are the peak users, whether a user account is used at many locations or only in a fixed location within the enterprise, etc will help in planning and optimizing network resources. If users are mobile they login from different locations hence use different IP addresses; hence the maximum number of users active in a subnet can be found. Knowing this may help in planning how many IP addresses have to be allocated for a subnet within a network. In order to answer similar questions we propose a method to profile the users based on their calling behavior. This Profiler collect and analyze SIP traffic to detect and differentiate registered VoIP users by their calling behavior and other characteristics like duration of call, location, etc. The proposed Profiler is a Deep Packet Inspection (DPI) engine for SIP packets and can identify user names, call ids, IP addresses used, etc which are subsequently used to generate the user profile. In particular we make following specific contributions in this paper.

1) We propose VoIP Profiler which is a Deep Packet Inspection (DPI) engine for parsing and extracting fields of interest from SIP packets pertaining to VoIP calls.
2) We define different types of users based on their calling patterns and other behavior we chose to monitor.
3) We define novel metrics for which statistics are calculated for each user and we subsequently use one or more of these metrics to identify a particular type of user.
4) We experiment with a large dataset containing approximately 1.4 million SIP packets and successfully identify various types of users.

Rest of this paper is organized as follows: In Section II we describe the steps involved in a VoIP call setup using SIP as a signaling protocol. In Section III, we describe related prior work. Our proposed Profiler is explained in Section IV. We describe the experiments done to evaluate the proposed method in Section V. Finally we conclude this paper in Section VI.

II. SIP Overview

SIP has a distributed architecture. It includes the entities as shown in Figure 1.

- **User agent:** These are VoIP phones with a valid URI (user name used by a user). Multimedia sessions are setup and terminated between user agents.

978-1-5090-0990-9/16 $31.00 © 2016 IEEE
DOI 10.1109/ARES.2016.19

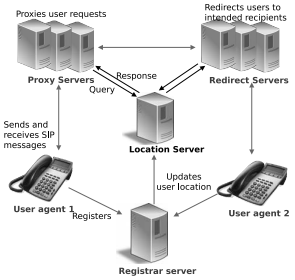

Fig. 1: VoIP Building Blocks

- **Registrar server:** User agents register with a registrar server when they connect to network and also update it periodically.
- **Location server:** Store different locations of user agents which are identified by their IP address. There can be more than one location associated with a user.
- **Proxy server:** Forward the connection requests to the intended recipients on behalf of user agents.
- **Redirect server:** If a user has more than one location, redirect server helps to fork a connection request to different addresses.

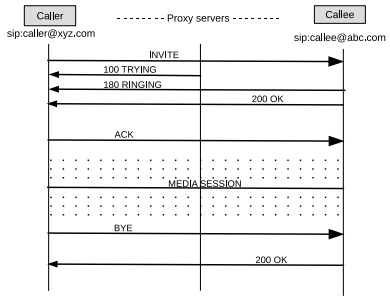

Fig. 2: VoIP Call Setup Suing SIP

Figure 2 shows an example call setup using SIP signaling. The call setup begins with an INVITE from the caller. SIP proxy server receives this INVITE and immediately sends a 100 (TRYING) message to caller. When the request reaches the callee, it starts ringing and it indicates the RINGING status in a 180 response. If the callee accepts the call, callee's phone sends a 200 (OK) response, otherwise an error response would be sent. Caller's phone acknowledges it by sending an ACK. This completes the three-way handshake INVITE/200/ACK used to establish SIP sessions. After this, media session runs. The media session uses Real time Transmission Protocol (RTP) after negotiating some parameters of media transmission and capabilities using Session Description Protocol (SDP). At the end of call, either of two user agents can send a BYE request to end the call. The other user agent responds to it with a 200 (OK) response.

III. Review of literature

In this section we describe relevant prior art for SIP based traffic monitoring and profiling. Traffic monitoring is mainly done for detecting different types of SIP attacks like malformed messages [23], flooding attacks [22], [27], [26], [7], colluding attacks [4], etc. Profiling is mainly done to understand the SIP traffic at the edge of a network. We discuss these two types in the following two subsections.

A. Traffic Monitoring for Anomaly Detection

There are a number of anomaly detection techniques which use statistics of various SIP messages to detect flooding and other attacks.

1) Flooding Attack Detection: Nassar et al. [17], [18] used Support Vector Machines to classify interval summaries as normal or attack cases to detect flooding attacks. Three types of features generated from SIP traffic statistics, SIP server logs and call activities are used for this purpose. Some works [22], [27], [26] have proposed to use hellinger distance (HD) to measure aberration between normal and different flooding attack traffic distributions. Authors of [15] propose a method for monitoring and detecting SIP-based VoIP network threats such as SIP flooding attack, RTP flooding attack and SIP scan based on Cisco NetFlow Version 9. Authors of [10] proposes to identify self similar traffic using Levenshteins distance and k-means clustering to filter anamolous messages assuming normal messages will be similar and anomalous messages are dissimilar.

2) SPIT Attack Detection: Anomaly detection method has also been used for Spam over Internet Telephony (SPIT) detection. Network traffic patterns of normal and Mass Call Spam (MCS) Attacks are compared in [11]. Patterns include statistical analysis of Message volume, Entropy of source UID, Poisson process analysis and INVITE interval analysis to detect abnormal traffic behavior by MCS Attacks. Each of these analysis methods uses a threshold to detect abnormal charecteristics by MCS. Improving performance of detection system and minimizing false positives is studied in [16]. Huang et al. [9] does voice activity analysis and detect SPIT callers using support vector machine. Guang-Yu et. al. [8] uses multi-layered detection and prevention system to detect SPIT calls.

B. SIP Traffic Profiling

There have been some attempts to understand the SIP traffic characteristics at the network infrastructure level. Kang et al. [13] proposed to profile the SIP traffic by passively monitoring the network. Profiling is done at different levels to identify SIP servers (IP address), Proxy Servers and different message type distributions. Stanek et al [24] studied the impact of Network Address Translators on SIP servers and also the geographical distribution of calls by analyzing traffic from an open source SIP server. A method of SIP traffic profiling at the network level, host level and proxy server level is proposed in [21]. However the profiling parameters and objectives are different.

IV. Proposed Work

In this section, we describe VoIP Profiler, the proposed method for analyzing SIP traffic and user behavior profiling.

This profiling is aimed at identifying different users and account types. In the first place we define a set of parameters which we call "call activity parameters" each measuring different types of messages and status of calls. Subsequently using these parameters we classify users and their behaviors.

A. Call Activity Parameters

These parameters measure the statistics for calling patterns, success rate of calls, etc for a particular type of user (identified by a user account or URI).

I) *Successful Call Rate*: This is the ratio of successful calls made by a caller to the total number of calls initiated by him/her. This is measured with α as given in Equation 1.

$$\alpha = \frac{\text{Successful Calls by User } u}{\text{Total Calls by User } u} \tag{1}$$

In Equation 1, the value α lies in the range of 0 to 1. If α is 0 then it indicates no successful calls are made by the user u and when it is 1 then all calls made by this user are successful.

II) *Talk-Time Per Day*: This defines the fraction of time user u is spending on call every day and is measured in β as shown in Equation 2.

$$\beta = \frac{\text{Total Talktime of User } u \text{ in a Day in Hours}}{\text{24 Hours}} \tag{2}$$

Here $\beta \in [0, 1]$. If β is 0 then user did not call anyone and if β is greater than 0 indicates fraction of time he/she is spending on call in a day. We measure it as an average of each day's call duration in the entire time of profiling.

III) *IP Address Diversity*: This is the number of different IP addresses used by a user. The value of γ in the Equation 3 is a natural number.

$$\gamma = \text{Total Number of IP Addresses Used by User } u \tag{3}$$

IV) *Average Talk-time Per Call*: This measures the average duration of each call for a user u. The parameter τ in Equation 4 is defined as a ratio of total talk time by the user to the total number of calls made by him/her. The value of $\tau \geq 0$.

$$\tau = \frac{\text{Total Talktime by User } u}{\text{Total Successful Calls Made by User } u} \tag{4}$$

V) *User Role in the Call*: This is the ratio of number of calls a user u makes to the number of calls she receives. This is measured by ρ as in Equation 5. If ρ is less than 1 then user is mostly a callee i.e. recipient of a call, if ρ is greater than 1 then user is mostly a caller i.e. makes a call and if ρ is equal to 1 then a user acts as both caller and callee in equal proportion.

$$\rho = \frac{\text{Number of Time User } u \text{ is a Caller in a Day}}{\text{Number of Times User } u \text{ is a Callee in a Day}} \tag{5}$$

VI) *Call Diversity*: This is a parameter indicating how many different users a particular user u calls. This includes other peer users within the same domain and outside.

TABLE I: Caller and Callee Relationship

Caller	Callee with number of calls
u_1	$u_2 = n_2$, $u_3 = n_3$, $u_4 = n_4$
u_2	$u_1 = n_1$, $u_3 = n_5$
u_3	$u_1 = n_6$
u_4	$u_2 = n_7$, $u_3 = n_8$

Call diversity of a user can be derived by maintaining a table of calls made by a user. Since every user's call details needs to be maintained this table will have those many entries as there are number of users in the domain. Sample call details of few users is shown in Table I. In the table, the first entry shows user u_1 is the caller and she called to three other users u_2, u_3 and u_4 with number of calls as n_2, n_3 and n_4 respectively.

VII) *Graceful Call Ending:* This parameter ψ indicates fraction of calls ended by a user (user ID) u with a BYE message which are graceful. A call is said to be gracefully ended if it has a preceding $INVITE$ message which is either sent by the user u or by a peer user who is part of this conversation (identified by a call ID).

$$\psi = \frac{\text{Total Calls Terminated By a User } u \text{ Gracefully}}{\text{Total BYE Messages Generated by the User } u} \tag{6}$$

B. Types of Users

As our aim is to do behavior profiling and subsequently classify the user based on their calling behavior we describe different types of users in this subsection. We identify several types of users based on their calling behavior. We also describe how using one or more of previously defined parameters we can identify different types of users.

I) *Based on Duration of Call*

1) Short Duration Callers: Callers who usually call for short duration of time are termed as short duration callers. These callers, in general consume little network bandwidth as their call duration is less.
2) Long Duration Callers: These callers make calls for very long duration of time. These callers use more bandwidth as compared to both Short Duration and Medium Duration callers.
3) Medium Duration Callers: These are the callers whose call duration is for a longer duration of time than the short duration callers and for lesser duration than long duration callers.

As describe in previous subsection τ is the average call duration and this can be used to identify above 3 types of callers. We can make use of appropriate thresholds on τ to classify users into one of these types.

II) Type of Callers: Based on type of calls made by a user u he/she can be one of the following type.

1) SPIT Callers: Its a general perception that, SPIT callers make unwanted calls and often calls are generated from a recorded media file (message) for marketing purpose or advertisement. Sometime these calls are also made by users in call center for promotional purposes of products. The callers or user ids which are used for this purposes are known as SPIT callers.
2) Non SPIT Callers: These callers or user IDs are those which engage in general conversation and not SPIT callers.

SPIT callers usually play recorded media files and many users (receivers) consider them as annoying and are likely to disconnect the calls prematurely. Hence the average call duration of such callers will be less. Also SPIT callers try calling many users in short period of time hence mainly act as callers in all these calls. Further many receivers notice the call ID used for such calls and are likely to reject the call this makes success rate of calls very low. We consider these three observations about SPIT calls and detect the SPIT user IDs using parameters α, ρ and τ. For a SPIT caller α and τ will be small while ρ will be high.

III) Call Location Diversity: Based on the location of user from where the call is originating a caller or user id can be classified into following two types.

1) Moving Callers: These are the user IDs or callers who log in from different locations. Users may log in from different sub-networks within the network.
2) Stable Callers: These are the user IDs or callers who log in from same locations.

Geographically moving users are likely to login from different locations and hence use different IP addresses. From the above list of parameters value of γ identify this. If the value of γ is 1 it means user is calling from same IP every-time and is a geographically stable caller. When $\gamma > 1$, then user has loged in from different places and hence may be a geographically moving caller. We can use a suitable threshold on γ for classifying user into either of these two categories.

IV) Frequency of Calls: Using the rate at which calls are made from a particular ID, an user ID or caller can be classified into following two categories.

1) INVITE Flooder: This is the user ID which is used to send multiple call requests targeted at a SIP server or a proxy server. Each call request results into an INVITE message thus there is a surge in number of such INVITES from that ID. A flooder can also do a flood against a particular user and keep that user busy. Such flooding accounts can be detected by again using a suitable threshold on number of INVITE messages generated by an user ID.
2) Normal User: A user ID which is not used as flooding source and call rate is within certain threshold is considered normal user.

SIP flooding can also be done using other types of SIP messages namely BYE, $REGISTER$ and a combination of above messages (including INVITE). Among these possible flooding cases, flooding with $INVITE$ messages is very common. Using the parameters α and β we can detect $INVITE$ flooding cases. It can be noticed that flooding cases does not complete the calls hence the value of α i.e success rate of call will be small and also talk time of caller will be less as these users have no interest in conversation.

VI) Graceful Call Ending: Based on the parameter ψ an user ID can be classified into either of two types.

1) Normal User: For any normal user, the parameter ψ will be balanced with no call being attempted to be ended without a prior establishment. In other words it strictly follows the VoIP call sequence of Figure 2.
2) BYE Flooder: For this user the balance between graceful and nongraceful call ending is skewed with random BYE messages being sent. These messages can terminate an ongoing call if the call ID of BYE messages match with any of the ongoing call and acts as a source of an attck overhelming the server as part of a DoS.

VI) Call Diversity: Based on to whom a caller makes calls user can be classified into following two types.

1) Diverse Caller: Callers who make calls to a wide range of users or call IDs.
2) Dedicated Caller: Callers who make calls to mostly specific users are called dedicated callers.

These type of callers can easily be detected by maintaining a list of peer call IDs of each caller. An example of such call list is shown in Table I. From this table we can notice that u_1 is the most diverse caller calling to 3 users u_2, u_3, u_4 with n_2, n_3, n_4 number of times whereas u_3 is the most dedicated user calling only to one user u_1 with n_6 number of calls.

V. Experimental Evaluation

In this section we describe the experiments done to evaluate the proposed user profiling method. We setup a testbed mimicking an enterprise network and generated VoIP traffic simulating user behavior. In the next two subsections we describe these two steps.

A. Experimental Setup

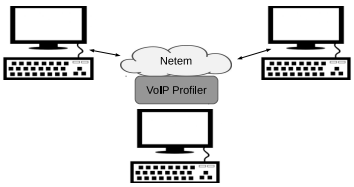

Fig. 3: Testbed

For the experiment purpose, we simulated an enterprise network by creating a testbed with 3 machines. This testbed is shown in Figure 3. All the three machines have Intel Core i5-4590 processor with clock rate of 3.30 GHz, 8GB of RAM and 1 GB of Intel dedicated premium graphics and run Ubuntu

12.04 LTS (Long Term Support). In one of the machine we setup Asterisk Server 11.18.0 [3] which acts as VoIP PBX[1]. The other two machines are used as client machines and generate VoIP calls. The client machines generate several VoIP calls using a modified version of VoIP bot [19]. Each bot program mimics a user and runs as independent process on a different UDP port. This bot program uses Jain SIP API to handle the SIP signaling and Java Media Framework (JMF) to generate RTP flow. In total we simulated 94 registered VoIP users in both the machines. We also installed a network emulator software called Netem [20] also in the server machine to simulate the Internet behavior.

B. Data Generation

I) *Generating Normal Calls*: Using the setup described above we generated and collected VoIP call data for a week. We used an open-source tool tcpdump [28] to capture packets from Asterisk server. The users (bots) are capable of registering at the registrar periodically as required. These bots made calls from client machines at random time intervals by choosing random bots, but in consideration with the real behavior. Each user made calls with a random delay of 1 minute to 180 minutes between successive calls. Call duration of each call is randomized between 10 seconds to 60 minutes. The dataset so generated is of 621.5 Mega Bytes and contain 1238289 packets. We wrote a Java program to analyze the collected traffic. This Java program parses each packet and uses JnetPcap [12] library. In this one week's data there were 12577 complete calls made by registered users.

II) *SPIT Calls*: For generating the SPIT calls, we use same setup used to generate Normal calls. There are couple of changes made. First is the Call duration is randomized between 10 seconds to 60 seconds instead of 10 to 60 minutes and second is these callers calls are received by other users with a low probability. We made 11 users (bots) to make short duration SPIT calls. As we are not determining SPIT calls using RTP just randomizing the call duration to a small range is OK. In total there are 1207 SPIT calls made during simulation.

III) *INVITE Flooding Calls*: We used 4 callers and 1 callee in a call scenario to generate INVITE flooding calls. Delay between two successive call trials by callers is now randomized between 1 second to 5 seconds. Once a pair of caller and callee enters in a call, other three callers try to connect to that callee with high trial rate. This generates huge number of INVITE request messages and simulates the behavior of INVITE flooding.

IV) *Stable and Moving Callers*: We created a program in which each bot gets unique IP address before making any call as a caller (through dhcp). For e.g bot1 has IP 192.168.2.1, bot2 has IP 192.168.2.2, bot30 has IP 192.168.2.30 and so on. But we selected 4 callers whose IP address is changed after certain number of calls. In this way, we simulated user using more than one IP address to behave like a moving caller.

V) *BYE Flooding Calls*: We generated BYE flooding cases by making 4 BOT users to generate BYE messages with random call IDs at rapid rate.

[1]A synonym for telephone exchange

VI) *Diverse and Dedicated Calls*: No different experiment is required for these callers. Once the data set is generated from above scenarios, we create a mapping of caller to unique callee. Those callers who called to most number of callee are the Diverse Callers whereas those called to least number of callee are Dedicated Callers.

TABLE II: Call Statistics

SN	Attribute	Calls
1	Complete Normal Calls	11539
2	Incomplete Normal Calls	712
3	Complete SPIT Calls	1173
4	Incomplete SPIT Calls	14120
5	Complete Calls by Flooders	19
6	Incomplete Calls by Flooders (Flooding)	27996
7	Total Complete Calls (1+3+5)	12731
8	Total Incomplete Calls (2+4+6)	42828

A summary of call details is shown in Table II. We use the dataset collected from these bot users and report various types of users detected using the parameters described previously. In order to do this we calculate all the parameters described in Section III A for each user from our experimental dataset. In the subsequent paragraphs we describe the findings of these calculations. Due to space limitation we do not show details of individual caller but a representation of each type is described. In most of the cases we select 5 users from each category and show the calculated call parameter statistics for these users. There are some categories of users in which detected users are less than 5, for such category we show details of all the users.

I) Identifying long duration, short duration and medium duration callers: As this category is based on average call duration of each user, we use suitable thresholds on call duration to categorize into one of three categories. As there are 12731 completed calls in the entire dataset. Figure 4. shows the histogram of call duration for each call. We use the time elapsed between the $INVITE$ and the corresponding BYE message as the duration of call. We can notice majority of calls have duration of less than 500 seconds and there are few long duration calls.

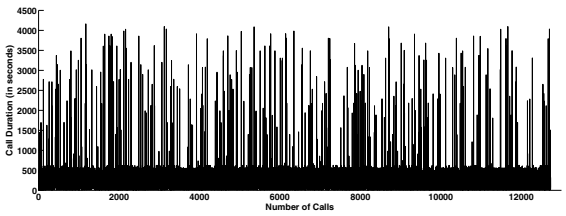

Fig. 4: Call Duration

We consider users whose average call duration is greater than 10 minutes (600 seconds) as long duration callers, those users whose call duration between 5 minutes (300 seconds) to 10 minutes (600 seconds) as medium duration callers and remaining users as short duration callers. Figure 5. shows the distribution of callers of each type found in the dataset from these threshold values. In particular we found 8 long duration callers, 37 medium duration callers and 35 short

TABLE III: Statistics for 5 Long Duration Callers

Caller Name	τ
bot43	688.38612
bot61	2379.18625
bot62	2550.02884
bot63	2137.25723
bot64	2458.67296

TABLE IV: Statistics for 5 Mid Duration Callers

Caller Name	τ
bot18	313.24245
bot24	308.96654
bot36	408.88238
bot41	585.07270
bot44	404.22538

duration callers. There are total 13 callers who made no calls. Table III, Table IV and Table V shows 5 representative users from long duration, medium duration and short duration callers respectively with parameters τ in seconds. A comparison of average talk time of each caller of all the three types of callers is shown Figure 6.

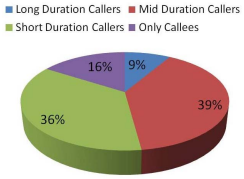

Fig. 5: Distribution of Callers based on Duration of Call

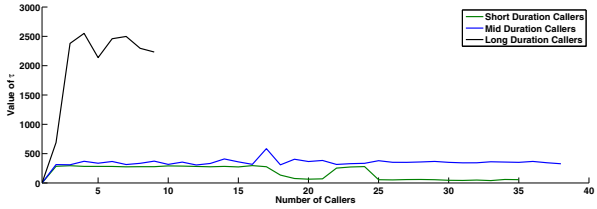

Fig. 6: Comparison of Average Talk-time (τ) of Callers

II) Detecting SPIT Callers: As described previously SPIT callers usually have very low average call duration, low success rate of calls and mostly act as callers. We use parameters τ, α and ρ to detect SPIT callers. Table VI shows 5 out of 11 detected SPIT callers from the dataset. In the table, it can be noticed that τ is lowest. It should also to be noticed that all these user IDs are majorly acted as a caller, therefore ρ is

TABLE V: Statistics for 5 Short Duration Callers

Caller Name	τ
bot23	282.06737
bot37	295.72963
bot45	134.29423
bot75	253.18883
bot77	272.54362

always greater than one for all the SPIT callers. This happens because users, usually do not give a call back to promotional calls until it is of their interest or they found an unknown missed call which may be from a SPIT caller. Similarly the call success rate of these callers is low compared with other normal users (shown in Table VI and Table VII)

TABLE VI: Statistics of 5 SPIT Callers

Caller Name	τ	α	ρ
bot50	30.54565	0.0919	8.7857
bot51	31.37514	0.0892	10.0000
bot55	28.35449	0.0846	7.4667
bot57	29.88865	0.0922	8.2000
bot59	30.07110	0.0908	13.4444

TABLE VII: Statistics of 5 Non SPIT Callers

Caller Name	τ	α	ρ
bot25	369.23053	0.1209	0.0748
bot30	334.54350	0.7143	0.7692
bot38	276.52861	0.6364	0.9130
bot41	585.07271	0.4118	0.8750
bot53	70.33555	0.6429	0.5000

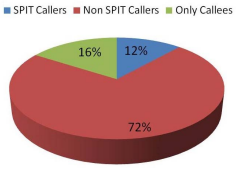

Fig. 7: Distribution of SPIT and Non SPIT Callers

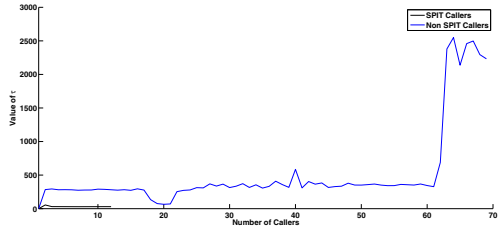

Fig. 8: Comparison of Average Talk-time (τ) of Callers

Figure 7. shows the distribution of SPIT and non SPIT callers identified from the dataset. Figure 8 shows a comparison of average talk time τ of user (average per call) of all SPIT and non SPIT users. Similarly the comparison of parameters α and ρ between SPIT and normal users are shown in Figure 9 and Figure 10 respectively.

III) *Detecting Stable and Moving Users*: We use parameter γ to detect stable and moving users. A snapshot of 5 highly moving users are shown in Table VIII. Users bot41, bot42, bot43 and bot44 are moving users as they have γ as 3, 2, 2, 4 respectively whereas all other users called from single IP and kept γ equals 1 and hence stable users. This means users having γ greater than one is moving and hence login from different machines. In our dataset we found 90 users who are stable users whereas 4 users who are moving.

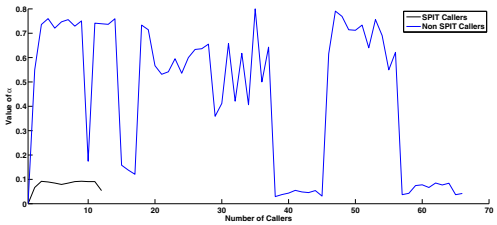

Fig. 9: Comparison of Call Success Rate (α) of Callers

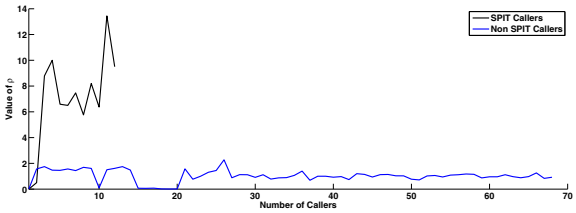

Fig. 10: Comparison of User Roll in the Call (ρ)

TABLE VIII: Statistics for Stable and Moving Users

Caller Name	γ	IP Addresses
bot41	3	192.168.2.131, 192.168.2.132, 192.168.2.133
bot42	2	192.168.2.141, 192.168.2.142
bot43	2	192.168.2.151, 192.168.2.152
bot44	4	192.168.2.161, 192.168.2.162, 192.168.2.163, 192.168.2.164
All other	1	-

Fig. 11: Distribution of Stable and Moving Callers

IV) *Detecting INVITE Flooding Users*: INVITE Flooders send multiple INVITES in which less are answered and hence makes α smallest among the callers. Table IX shows 4 detected flooding users with a threshold of 0.1 for α and β. Similarly Table X shows the normal users statistics. These are the users with the least value of α among the registered ones. It can also be noticed that the average call duration β for these callers is also low compared to normal users. Figure 12 shows the distribution of INVIITE flooders and other users and Figure 13 shows a comparison of call success rate for the two types of users.

TABLE IX: Statistics of Flooders

Caller Name	α	β
bot26	0.0013	0.00023
bot27	0.0009	0.00020
bot28	0.0007	0.00030

TABLE X: Statistics of 5 Normal Users (Non Flooders)

Caller Name	α	β
bot20	0.7389	0.03766
bot21	0.7365	0.03888
bot22	0.7594	0.03496
bot71	0.7692	0.01068
bot73	0.7121	0.01419

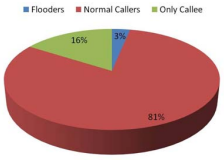

Fig. 12: Distribution of Flooders and Normal Callers

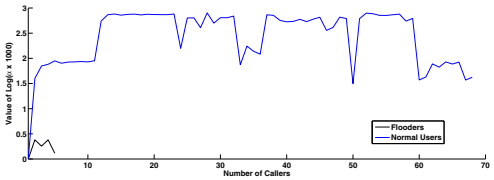

Fig. 13: Comparison of Successful Call Rate (α) of Callers

V) *Detecting BYE Flooding Callers*: We used parameter ψ to detect the user IDs who are sending BYE flooding messages. Table XI and Table XII shows the value of ψ for BYE flooding user agents and non flooding users. As we can see from the two tables the ψ value flooding users is very small and for non flooding users it is around 1. By setting a threshold of 0.1 on ψ BYE flooding users (user agents) can be detected. Figure 14 is the comparison graph of gracefull call termination ration between BYE flooder and normal user. Figure 15 shows the distribution of BYE flooders and normal users detected from the dataset. It should be noted that those users who are calee can also terminate the call hence there is no difference between caller and calee here.

TABLE XI: Statistics of BYE Flooders

User name	ψ
bot6	0.0492
bot7	0.0531
bot8	0.0520
bot9	0.0517

TABLE XII: Statistics of 5 Normal Users

User name	ψ
bot31	0.9905
bot32	0.9901
bot33	1.0000
bot34	1.0000
bot35	1.0000

VI) *Detecting Dedicated and Diverse Callers*: Table XIII shows the list of 3 users who are identified as dedicated callers.

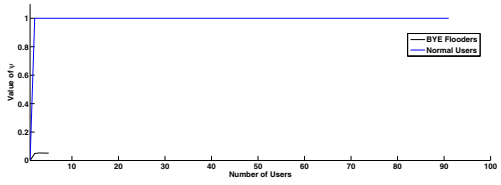

Fig. 14: BYE Flooder and Normal User Comparison based on Graceful Call Termination

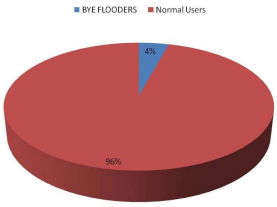

Fig. 15: Distribution of Callers based on Graceful Call Termination

Out of 94 users, bot46, bot47 and bot48 calls only to single callee.

TABLE XIII: Statistics of Top 3 Dedicated Callers

Caller Name	Number of Callee	Name of Callee
bot46	1	bot49
bot47	1	bot46
bot48	1	bot45

Similarly Table XIV shows the 3 most diverse callers who made calls to a large number of other users. In the dataset we found bot82, bot83 and bot86 are most diverse callers which called to at least 20 other users.

TABLE XIV: Statistics of Top 3 Diverse Callers

Caller Name	Number of Callee	Name of Callee
bot82	20	bot68 to bot88
bot83	25	bot68 to bot89 except bot76
bot86	32	bot68 to bot89 except bot73

VI. Conclusion

In this paper we presented VoIP Profiler a method for profiling user level behavior using VoIP traffic. We defined a set of parameters to measure for each user and subsequently used one or more of those parameters for classifying VoIP user types or user IDs into one or more category. We believe that the fine grained profiling done by VoIP Profiler helps in many ways including identifying attacks of different types, planning for infrastructure deployment and optimization and also in policy decision making in the organization. We experimented with a simulated network environment and correlated the classification of user to parameters on which it was based. We intend to experiment with different SIP attack types and correlate detection using different parameters in the future.

References

[1] RFC 3261. https://www.ietf.org/rfc/rfc3261.txt.

[2] R. Arora and R. Jain. Voice over ip: Protocols and standards. *Network Magazine, 23rd of November*, 1999.

[3] Asterisk. http://www.asterisk.org/.

[4] W. Conner and K. Nahrstedt. Protecting sip proxy servers from ringing-based denial-of-service attacks. In *ISM'08: In The tenth IEEE international symposium on multimedia*. IEEE, 2008.

[5] I. Dalgic and H. Fang. Comparison of h.323 and sip for ip telephony signaling. In *Photonics East'99*, volume 3845, pages 106–122, 1999.

[6] D. Geneiatakis, G. Kambourakis, C. Lambrinoudakis, T. Dagiuklas, and S. Gritzalis. A framework for protecting a sip-based infrastructure against malformed message attacks. *Computer Networks*, 51(10):2580–2593, 2007.

[7] D. Goliat and N. Hubballi. VoIPFD: Voice over ip flooding detection. In *NCC '16: Proceedings of the 22nd National Conference on Communication*, pages 1–6. IEEE, 2016.

[8] H. Guang-Yu, W. Ying-You, and Z. Hong. Spit detection and prevention method based on signal analysis. In *Convergence and Hybrid Information Technology, 2008. ICCIT'08. Third International Conference on*, volume 2, pages 631–638, 2008.

[9] H. Hai, Y. Hong-Tao, and F. Xiao-Lei. A spit detection method using voice activity analysis. In *Multimedia Information Networking and Security, 2009. MINES'09. International Conference on*, volume 2, pages 370–373, 2009.

[10] N. Hantehzadeh, A. Mehta, V. Gurbani, L. Gupta, T. Ho, and G. Wilathgamuwa. Statistical analysis of self-similar session initiation protocol (sip) messages for anomaly detection. In *NTMS'11: Proceedings of the 2011 IFIP International Conference on New Technologies, Mobility and Securi*, pages 1–5. IEEE, 2011.

[11] J. Heo, T. Kusumoto, E. Chen, and M. Itoh. A statistical analysis method for detecting mass call spam in sip-based voip service. In *APSITT '10: Proceedings of 8th Asia-Pacific Symposium on Information and Telecommunication Technologies*, pages 1–6. IEEE, 2010.

[12] JnetPcap. http://www.jnetpcap.com/.

[13] H. Kang, Z. Zhang, S. Ranjan, and A. Nucci. Sip-based voip traffic behavior profiling and its applications. In *MineNet '07: Proceedings of the 3rd annual ACM workshop on Mining network data*, pages 39–44, 2007.

[14] A. Keromytis. A comprehensive survey of voice over ip security research. *IEEE Communications SurveysS & Tutorials*, 14(2):514–537, 2012.

[15] C-Y. Lee, H k. Kim, K-H. Ko, J-W. Kim, and C. Jeong. A voip traffic monitoring system based on netflow v9. *International Journal of Advanced Science and Technology*, 4(1):1–8, 2009.

[16] F. Menna, R. Cigno, S. Niccolini, and S. Tartarelli. Simulation of spit filtering: Quantitative evaluation of parameter tuning. In *ICC'09: Proceedings of the 2009 IEEE International Conference on Communications*, pages 2162–2167. IEEE, 2009.

[17] M. Nassar, R. state, and O. Festor. Monitoring sip traffic using support vector machines. In *RAID '08: Proceedings of the 11th International Symposium on Recent Advances in Intrusion Detection*, pages 311–330, 2008.

[18] M. Nassar, R. state, and O. Festor. A framework for monitoring sip enterprise networks. In *NSS '10: Fourth International Conference on Network and System Security*, pages 1–8, 2010.

[19] Md. Nassar, O. Festor, et al. Labeled voip data-set for intrusion detection evaluation. In *Networked Services and Applications-Engineering, Control and Management*, pages 97–106. Springer, 2010.

[20] Netem. http://www.linuxfoundation.org/collaborate/workgroups/networking/netem.

[21] A. Nucci and S. Ranjan. Sip based voip traffic behavior profiling method.

[22] H. Sengar, H. Wang, D. Wijesekera, and S. Jajodia. Detecting voip floods using the hellinger distance. *IEEE Transactions on Parallel and Distributed Systems*, 19(6):794–805, 2008.

[23] D. Seo, H. Lee, and E. Nuwere. Sipad: Sipvoip anomaly detection using a stateful rule tree. *Computer Communications*, 36(5):562–574, 2013.

[24] J. Stanek, L. Kencl, and title = J. Kuthan.

[25] J. Tang, Y. Cheng, and Y. Hao. Detection and prevention of sip flooding attacks in voice over ip networks. In *INFOCOM '12: Proceedings of the 31st Annual IEEE International Conference on Computer Communications*, pages 1161–1169. IEEE, 2012.

[26] J. Tang, Y. Cheng, Y. Hao, and W. Song. Sip flooding attack detection with a multi-dimensional sketch design. *IEEE Transactions on Dependable and Secure Computing*, 11(6):582–595, 2014.

[27] J. Tang, Y. Cheng, and C. Zhou. Sketch-based sip flooding detection using hellinger distance. In *GLOBECOM'09: Proceedings of the 28th IEEE Conference on Global Telecommunications*, pages 3380–3385. IEEE, 2009.

[28] Tcpdump. http://www.tcpdump.org/.