Towards a Forensic Analysis for Multimedia Communication Services

Dimitris Geneiatakis and Angelos D. Keromytis

Department of Computer Science, Columbia University
1214 Amsterdam Avenue, M.C. 0401
New York, 10027-7003 USA
{dgen,angelos}@cs.columbia.edu

Abstract— **No matter how robust the employed security mechanisms are malicious users or attackers will always find a way to bypass them. In addition, National Institute of Security and Technology mentions *"In conjunction with appropriate tools & procedures, audit trail can assist in detecting security violation and flaws in applications"*. Until now, in Multimedia Communication Services (MCS), such as Voice over IP, audit trails are not utilized in security audits due to (a) the lack of the appropriate analysis tools and (b) privacy restrictions. In this paper we report on the analysis of MCS audit trail by employing a novel method for identifying "uncommon" traffic indicating non normal behaviour that does not violate users' privacy. We rely on entropy theory and the notion of "itself information" to quantify the randomness of specific message segments, and we also introduce the term "actual itself information" for the assessment of entire message randomness. To protect users' privacy we hash audit trail's data. For evaluating the applicability of our proposed method we utilize an audit trail of a real MCS provider published by honeypot project. Initial outcomes show the feasibility of employing such a method to recognize "uncommon" traffic, recorded in MCS audit trail.**

Keywords— **Multimedia Communication Services, Voice over IP, Digital Forensic, Entropy**

I. INTRODUCTION

Legacy telecommunication services provided by Public Switch Telephone Networks (PSTN) focus mainly on voice provision. In PSTNs, security, availability and reliability rely on a closed network architecture, while effective for voice provision is considered an obstacle for the employment of new advanced multimedia services. On the other hand, Internet offers to telecommunication providers a variety of chances to develop and employ new added value services at low cost. Providers such as *skype (www.skype.com), iptel (www.iptel.org)* are offering web based conference rooms, namely Multimedia Communication Services (MCS), not only for exchanging voice but also video, text and files at no cost for Internet connected users.

Although users experience advantages from MCS, the main drawbacks are vulnerabilities, threats and attacks introduced by the Internet's open architecture. Several researches [1-3] indentify and report security flaws like fuzzing, flooding and man in the middle (MitM) attacks. Simultaneously, in order to shield MCS against these threats different approaches have been proposed [4-6] by the research community, while MCS providers and product vendors suggest the employment of specific best practices [7],[8]. However, no matter how strong the employed security mechanisms and countermeasures are, malicious users will always find a method to bypass them.

Besides, administrators would like to examine the security robustness of their information systems to ensure that systems are free from attacks. For this purpose, administrators are able to utilize different techniques and methods to evaluate system security. For example, to identify vulnerabilities in the provided services administrators can launch specific attack scenarios using attack tools such as *nessus (www.nessus.org), Retina (www.eeye.com)* etc, or they might assess the entire system security by applying an audit methodology such as CRAMM [9], COBIT [10]. At this point it should be mentioned that security evaluation approaches, in most cases, do not take into account the existing audit trails, mainly due to the lack of appropriate tools for examining these audit trails. Consequently, it might be mistakenly assumed that the provided services are secure, while in reality they are vulnerable to existing security flaws (e.g resource consumption or other type of denial of service attacks) that have not been identified or recognized yet, but "reside" in the provided service. Even, the National Institute of Standards and Technology (NIST) in [11] mentions "*In conjunction with appropriate tools & procedures, audit trail can assist in detecting security violation and flaws in applications*".

For these reasons, it is of vital importance to identify "uncommon" traffic that bypass the existing security countermeasures and affects service's availability, and security by examining MCS audit trail in order to enhance provided service security. Additionally, whenever such an analysis is accomplished should take into consideration users' privacy issues as audit trails include among the others personal information. In this paper we report the analysis of MCS audit trail by using a novel method for identifying "uncommon" traffic indicating a security incident against MCS. By the term "uncommon" traffic we refer to well formed messages utilized by a malicious user to launch a Denial of Service (DoS) attack. The proposed method is based on information entropy theory [12]. For identifying "uncommon" traffic we rely on the "itself information" of the entropy theory and introduce a new metric namely *"actual information"* to evaluate entire's message randomness for detecting "uncommon" traffic. We utilize the hash form of raw data to protect legitimates' users privacy. Note that in our analysis we assume that the only available information to security examiners is the audit trail.

Even though in this work we focus on MCS services utilize Session Initiation Protocol (SIP) [13], as SIP has been adopted as the signalling protocol to handle multimedia sessions in Next Generation Networks (NGNs), the proposed method can also be employed in other multimedia and text based protocols.

To evaluate our method we utilize an MCS SIP based service provider audit trail file that includes various types of attacks according to the "*honeypot project"* (www.honeypot.org). Our preliminary outcomes demonstrate that the proposed method is able to identify particular "uncommon" messages that have been involved in an attack attempt, without violating legal users' privacy.

The reminder of this paper is organized as follows. In Section II we briefly describe the MCS infrastructure focusing on SIP. In Section III we introduce the proposed MCS security analysis model. In Section IV, we present some preliminary results in terms of effectiveness and in Section V we discuss the related work. Finally, in Section VI we conclude our work and give pointers for future work.

II. Multimedia Communication Services: Architecture

A. MCS General Architecture

MCS services are designed with Internet architecture in mind. As a result MCS has been developed as an application layer service in the Internet architecture stack. At network layer MCS utilize Internet Protocol (IP) [14], while at transport either User Datagram Protocol (UDP) [15] or Transport Control Protocol (TCP) [16] depending on the type of the provided services. At application layer MCS does not require only the exploitation of existing protocols like Domain Name System (DNS) [17] but also multimedia aware protocols. Particularly, for multimedia session handling, different protocols have been proposed, such as SIP [13], H.323 [18], MGCP [19], however for Internet based MCS, SIP is considered the predominant protocol. Regarding media transport protocols Real Time Protocol (RTP) [20] is utilized.

B. Session Initiation Protocol & MCS

SIP is an application layer protocol on Internet architecture responsible for handling multimedia sessions. A, SIP network infrastructure is consisted of the following logical network entities:

- *Registrar*: proceeds users' registration and store their current position in order to forward them new incoming requests and responses.
- *Proxy*: locates the requested resource and provided it to the end user. Note that the requested resource could be either multimedia content or an MCS end-user, depending on the type of the provided service.
- *User Agent*: correspond to end – users entities that acting on behalf of them by generating the appropriate request or responding to an incoming one.

Figure 1 illustrates the general network architecture of an MCS SIP based service architecture, as well as, the message flow followed whenever a user would like to exploit a service. SIP messages format follows a text based structure (see Figure 2) similar to HTTP message structure [21]. A more detailed analysis for SIP could be found in [13].

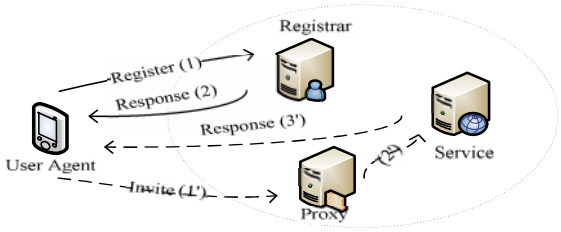

Figure 1. An MCS SIP Based Network Architecture

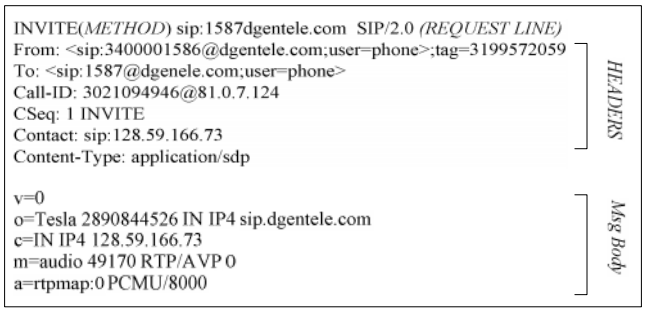

Figure 2. An Example of SIP Message Structure

III. An Entropy Based Analysis Model

A. General Description

It is beyond doubt that the appropriate analysis of audit trails (log files) will exhibit the effectiveness of the employed security mechanism, as an audit trail records every activity. Taking into account also that malicious users in MCS services utilize very similar messages (e.g flooding, signalling attacks) in order to accomplish their attacks' goals, we use entropy theory and the notion of "itself information" to identify message redundancies correspond uncommon behaviour by analysing an audit trail. Before applying an analysis on the raw data we hash them to protect users' privacy. The proposed solution is consisted of two basic modules:

- *Hash:* Users' privacy is protected by hashing the information included in MCS audit trail. As a result, no one entity is able to deduce users' specific preferences by processing the hashed file instead of the actual audit trail.
- *Entropy:* This module identifies messages that have been used in attack incidents against MCS, and can be utilized to recognize the specific characteristic of the attack.

Figure 3 depicts the functional diagram of the proposed approach.

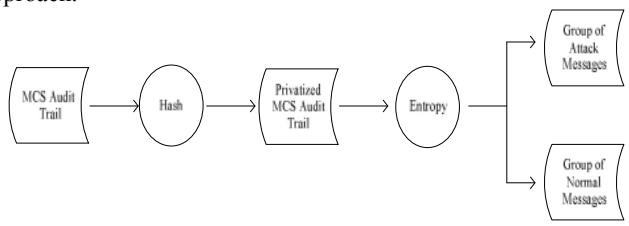

Figure 3. The Functional Scheme of the Proposed Analysis Model

B. Description of the Proposed Model

1) Itself Information & Entropy

According to Shanon's mathematical theory of communication [12], entropy is *a measure of how uncertain we are of an outcome*. Specifically, considering that a symbol $A(i)$ in a specific set S has probability $P_A(i)$, then the "itself information" (included in this symbol) is by definition:

$$I_A(i) = -log_b\big(P_A(i)\big) \quad (1)$$

The average of "itself information" connected to the set S is called entropy and is computed using the following formula:

$$H(S) = -\sum_{i}^{n} P_A(i) \times log_b P_A(i) \quad (2)$$

The entropy of a source set S maximizes when all $P_A(i)$ are equal ($P_A(i) = 1/n$). This means that the uncertainty of the outcome is maximized, while the redundancy in set S is minimized. With respect to "self information" this fact indicates that all messages (or alternatively symbols) contain the same amount of information. Note that the greater the probability of a specific message the less information is included in it. Furthermore, in case where two symbols are independent of each other then the "itself information" and the entropy are calculated using the following formulas, correspondingly:

$$I(A, B) = I(A) + I(B) \quad (3)$$

$$H(A, B) = H(A) + H(B) \quad (4)$$

2) Our Context

The main purpose of this work is not only to identify specific set of messages as malicious, but also to distinguish the legitimate traffic from malicious one. Furthermore, given that signalling protocols in MCS like SIP are text based with high level of syntax freedom, it is not possible to predetermine whether a group of messages are a part of a flooding attack or any other type of attack. Note that the messages of resource consumption flooding attacks and other types of DoS such as signalling attacks are based on well-formed messages that cannot be distinguished from legitimate ones. The only difference from legitimate messages is the amount of similar symbols included in the malicious messages.

Within this context, to compute the "itself information" of each independent message existing in an SIP based MCS audit trail, and the corresponding entropy values we define the following sets:

- *S1:* corresponds to the entire SIP message.
- *S2:* corresponds to the FROM header of a SIP message.
- *S3:* corresponds to the TO header of a SIP message.
- *S4:* corresponds to the Call-ID header of a SIP message.
- *S5:* corresponds to the *First-Line* (requested resource) of a SIP message.

The choice of these sets reflects the different types of SIP messages that a malicious user could craft in order to launch a resource consumption attack. A malicious user, for example, could use exact the same message (*S1* set) or may craft different messages by modifying specific segments of MCS message such as FROM, TO, Call-ID, headers or even the First Line (sets *S2* to *S5*) depending on the goal of the attack. An "artificial" sample of these sets illustrates Table 1, where each cell corresponds to a specific symbol of the set.

TABLE 1. EXAMPLE OF DEFINED SETS

	Defined Sets				
	S1	*S2*	*S3*	*S4*	*S5*
Symbols	S1a1	S2b1	S3c1	S4d1	S5e1
	S1a2	S2b1	S3c2	S4d2	S5e1
	S1a3	S2b1	S3c3	S4d3	S5e1
	S1a4	S2b1	S3c4	S4d4	S5e2
	S1a5	S2b1	S3c4	S4d5	S5e2
	S1a6	S2b1	S3c4	S4d6	S5e2
	S1a7	S2b1	S3c5	S4d7	S5e2
	S1a8	S2b1	S3c6	S4d8	S5e2
	S1a9	S2b2	S3c6	S4d9	S5e2

According to entropy theory symbol redundancy indicate lower entropy values. This means that some symbols in the set have greater frequency corresponding to less "itself information" compared to other symbols coexist in the examined set. So, from our point of view under the ideal case in an MCS service should not exist message redundancies in the incoming requests (SIP messages), except those occur due to retransmission.

In our model a message is characterized as "uncommon" if the itself information of a message is below a specific threshold td_1 or the *distance* from the ideal case is greater than a threshold td_2, during a time window *tw*. By *distance* we refer to the difference between the maximum value and message's current "itself information" value. Exact the same case is for entropy values in order to identify which specific set a malicious user exploit during an attack. All thresholds are parameters that should be defined by the MCS service provider depending on the specific characteristics. A detailed analysis of how to compute the appropriate thresholds is out of the scope of this paper.

Furthermore, instead of examining each set "alone" we define the *actual* "itself information" of the message as well as the *actual* entropy of a set, taking into account that in our model a message is consisted of *S2,S3,S4 and S5* sets. In that case the "itself information" and the entropy will be

$$I(S2, S3, S4, S5) = I(S2) + I(S3) + I(S4) + I(S5) \quad (5)$$

$$H(S2, S3, S4, S5) = H(S2) + H(S3) + H(S4) + H(S5) \quad (6)$$

3) Deployment Example

To demonstrate the proposed model usability we provide herein a specific example. A real case scenario is presented in Section IV. Particularly, considering that Table's 1 data set correspond to a privatized MCS audit trail raw data we compute:

- the "itself information" for every part of the message
- the *actual* "itself information" of the message
- the entropy of each of these sets and
- the *actual* entropy of each of these sets

by applying the equations (1), (2), (5) and (6) correspondingly. For example, the probability for each symbol of *S1* set is $P(i) = 1/9$, the "itself information" for each symbol of *S1* set is $I(i) = -\log(P(i))$, $i = S1a_i$, and *S1* set's entropy value is $H(S1) = \log n, n = 9$. Similarly, we compute the corresponding values for the remaining sets (*S2*-*S5*) (see Table 2 & 3).

TABLE 2. EXAMPLE OF ITSELF INFORMATION

	S1	*S2*	*S3*	*S4*	*S5*	*Actual*
Itself Information	3.17	0.17	3.17	3.17	1.58	8.09
	3.17	0.17	3.17	3.17	1.58	8.09
	3.17	0.17	3.17	3.17	1.58	8.09
	3.17	0.17	1.58	3.17	0.58	5.50
	3.17	0.17	1.58	3.17	0.58	5.50
	3.17	0.17	1.58	3.17	0.58	5.50
	3.17	0.17	3.17	3.17	0.58	5.50
	3.17	0.17	2.16	3.17	0.58	6.08
	3.17	3.17	2.16	3.17	0.58	6.08

TABLE 3. EXAMPLE OF SETS ENTROPY

	S1	*S2*	*S3*	*S4*	*S5*	*actual*
Entropy	3.17	0.17	2.29	3.17	0.91	6.54
Max	3.17	3.17	3.17	3.17	3.17	12.68

Although the raw data of Table 1 is very limited and artificial, by examining Table 2 & 3 can be deduced that set S2 and S5 includes more redundant symbols compared to the other sets. In this point, it should be noted that while the "itself information" (see Table 1) of *S1* illustrates that all incoming request are different, (this is right because every new incoming message, except retransmissions, should be different), the *actual* "itself information" depicts that the messages 4-7 include more redundant symbols compared to other messages, which correspond to the actual symbol redundancy included in these messages. This is the reason why we define the *actual* "itself information" metric. Additionally, the difference between the *actual* "itself information" of these specific messages and the corresponding maximum case is approximately 57% giving a clear indication of symbol redundancy in messages included in the audit trail.

By examining the Table 3 we can identify the set with lower entropy to deduce the type of attack launched against the provided MCS service. The sets *S2* and *S5* not only have the lower entropy values compared to the other sets but also the difference between maximum entropy value (ideal case) *S2* and *S5* entropy values (defined as distance in our context) are approximately to 95% and 72% correspondingly, meaning that the attacker has utilized SIP messages with the similar *S2* (From header) and *S5* (First Line) symbols. We have not defined any specific threshold because the audit trail in this case is "artificial", however, even in this case by the employment of the proposed solution we can deduce the existence of "uncommon" traffic.

Considering all the above, the proposed solution could be utilized to identify "uncommon" traffic indicating a DoS attack traffic by analysing only an audit trail. Even in case that we have not define specific thresholds service by observing only the "itself information" and entropy values we have an indication (or not) of a DoS attack. In this point one might argue that a basic limitation of the proposed model is the case of "complete" distributed attack in which all the messages will be different and consequently the "itself information" and entropy will be maximized. However, this case is similar to mother's day effect, in which a numerous of legal request overloads a service. In the following section we present a real case to confirm our initial outcomes.

IV. EVALUATION & QUALITATIVE ANALYSIS BASED ON REAL DATA

To evaluate the proposed method in terms of effectiveness, instead of using artificial data (Section III.B.3) we use an MSC audit trail file of a real provider, downloaded from the *honeypot project* (www.honeypotproject.org). This audit trail corresponds to five minute network traffic and contains 4266 signalling messages. To the best of our knowledge this is the only published raw data of an MCS provider.

Based on our approach, at the first step is preserved users' privacy, by hashing the raw data. So, the examiner is not able to identify real users' identities and their preferences. Moreover, by such a method a provider could publish its log data in order to "prove" its security level, without affecting users' privacy and violating any privacy legislation. At the next step, we compute the "itself information" for each symbol and the corresponding sets entropy. At the final step is identified whether a specific message was a part of a DoS attack as well as can be deduced the type of the attack.

Figures 4 to 8 illustrate the "itself Information" for SIP Message, From, To, First Line and the *actual* "itself information" correspondingly, while the entropy values of the sets S1-S5 are presented in Table 4.

TABLE 4. REAL CASE ENTROPY VALUES

	S1	*S2*	*S3*	*S4*	*S5*
Entropy	12.06	8.75	8.61	12.06	8.07
Max	12.06	12.06	12.06	12.06	12.06

Looking first at entropy values is clear that symbol redundancy in set S2, S3 and S5 is 27%, 28% and 33% respectively (the lower the entropy the greater the redundancy). This fact represents "uncommon" traffic that either corresponds to malicious user's "different" attempts to cause a DoS or to misconfigured user agents.

These sets of traffic could be considered normal only if the specific MCS provider has defined an entropy distance threshold greater than 30% for a time window 5 minutes. As we are not able to infer safely the existence of "uncommon" traffic without having particular entropy thresholds, we should examine the "itself information" as well. Similar to the example presented in Section 3, *S1* set's "itself information" shows that all the incoming requests correspond to different SIP messages (all the requests are different in the examined file), without any degree of similarity (see Figure 4). However, by examining individually the "itself information" for sets *S2, S3* and *S5* (see Figure 5, 6 and 7) we can validate the existence of similar symbols in these sets. For example the "itself

information" of messages with id greater than 3200 has decreased at the value of 2, where the value of ideal case is 12.06. Additionally, the *distance* between the ideal case (maximum "itself information") and the corresponding "itself information" values of set S1 for the messages with id number between 3200 – 4000 is almost 90%, which is clear indication of non-normal behaviour corresponding to "uncommon" traffic. Note, that we have not used any specific threshold because of lack of previous network traffic for this specific MCS provider. At this end as we have mentioned earlier the *actual* "itself information" correspond to the real symbol redundancy included in the examined data (see Figure 8). Consequently, instead of looking on different parts of the messages we suggest the usage of *actual itself information*.

To validate our outcomes we examine the real audit trail first for the message with lower "itself information". The results shows that the set of messages with lower "itself information" was different registration (SIP REGISTER) attempts by malicious users in order to identify legitimate users' passwords. For accuracy reasons we scrutinized the audit trail for the messages with higher "itself information", which actually was different scan attempts utilizing SIP OPTIONS method. The reason that our solution is not able to identify these messages as "uncommon" traffic is because these types of messages do not include redundant symbols. Actually, this is the way that a legitimate user's agent operates to identify other users' capabilities.

The outcomes demonstrate that the proposed solution can effectively identify specific messages corresponding to "uncommon" traffic. In any case in order to have a clear indication of attack traffic against a specific provider is suggested the employment of the appropriate entropy and "itself information" thresholds. On top of that the definition of *"actual"* "itself information" and entropy reflect correctly to the symbol redundancy existing in a set of messages. All in all our analysis can be utilized to evaluate and enhance MCS security level as it is able to identify "uncommon" traffic examining only the audit trail.

V. Related Work

Various researches [22-26] have been accomplished to identify resource consumption attacks as a part of network intrusion detection systems (IDS), however, very few focus on the analysis of MCS audit trail to identify and distinguish "uncommon" traffic. Herein we refer only to the research works that have some similar characteristics [22][26] to our proposed model. Particularly, in [22] it is proposed an entropy based mechanism to detect Spoofed Denial of Service attacks by monitoring the distribution of destination/source IP addresses; an alarm is raised if the entropy of specific IP address set is less than a predefined threshold value. A similar approach is used in [26] as well. These types of methods can be utilized in MCS services to analyse audit trails or real data traffic only on the IP level as do not take into consideration application layer information. In our solution we are based on entropy model as well, but we follow a difference approach. We do not focus only on entropy, but also we utilize the notion of "itself information" in order to identify specific "uncommon" (well formed) messages. In addition we should mention that in our approach we consider that the only available information to security examiners are the audit trails. The only published work that focus on forensic analysis in MCS is presented in [27], which use predetermine attack patterns to identify malicious events, by combining information from different network components.

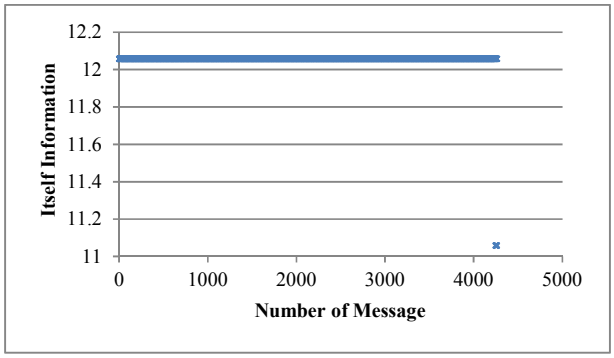

Figure 4. SIP Message (S1 set) Itself Information

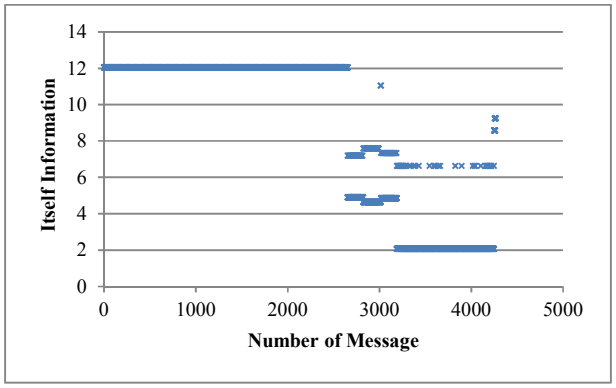

Figure 5. FROM Header (S2 set) Itself Information

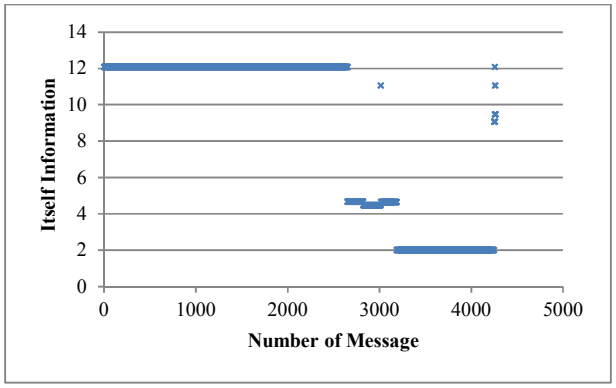

Figure 6. To Header (S3 set) Itself Information

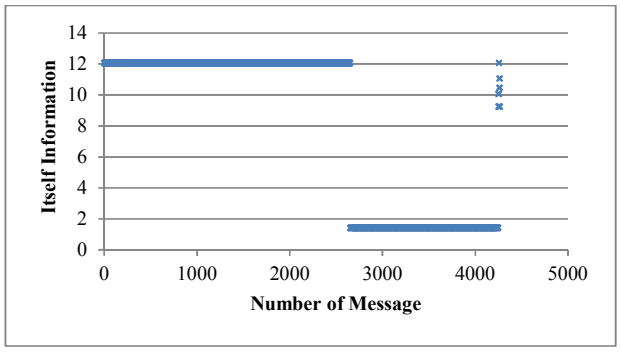

Figure 7. SIP First Line (S5 set) Itself Information

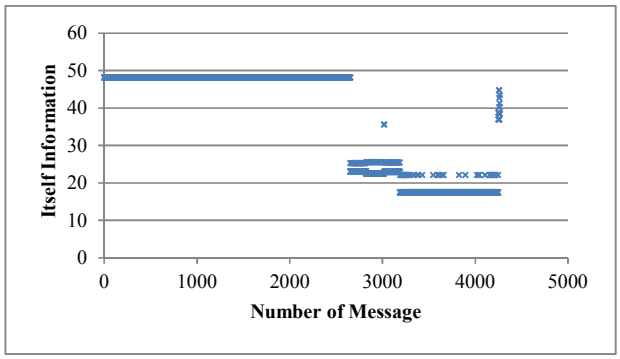

Figure 8. *Actual* Itself Information

VI. Conclusions & Future Work

MCS providers do not utilize the audit trails during a security audit due to (a) the lack of the appropriate tools, and (b) privacy restrictions. Under this context, we propose a method that by pass privacy restriction, while based on our initial outcomes it provides an effective technique to identify uncommon traffic. Particularly, we have exploit the "itself information" and entropy to measure message randomness to identify abnormal messages. Our initial results demonstrate that we can reveal abnormal behavior that residing in an MCS audit trail file. However, more extended tests are required in order to validate and confirm the outcomes derived in this work. Additionally, we are considering for applying the proposed method as an intrusion prevention mechanism to distinguish "uncommon" traffic in order to enhance MCS security level.

Acknowledgement

This work was supported by the NSF through Grant CNS-09-14845, by ONR through MURI Contract N00014-07-1-0907, and by AFOSR through MURI Contract FA9550-07-1-0527. Any opinions, findings, conclusions or recommendations expressed herein are those of the authors, and do not necessarily reflect those of the US Government, ONR, AFOSR, or the NSF.

References

[1] A.D. Keromytis, "Voice-over-IP Security: Research and Practice," *IEEE Security and Privacy*, vol. 8, 2010, pp. 76-78.

[2] D. Geneiatakis, T. Dagiuklas, G. Kambourakis, C. Lambrinoudakis, S. Gritzalis, K. Ehlert, and D. Sisalem, "Survey of security vulnerabilities in session initiation protocol," *Communications Surveys & Tutorials, IEEE*, vol. 8, 2006, pp. 68-81.

[3] D. Sisalem, J. Kuthan, and S. Ehlert, "Denial of service attacks targeting a SIP VoIP infrastructure: attack scenarios and prevention mechanisms," *Network, IEEE*, vol. 20, 2006, pp. 26-31.

[4] 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*, vol. 51, Jul. 2007, pp. 2580-2593.

[5] D. Geneiatakis, N. Vrakas, and C. Lambrinoudakis, "Utilizing bloom filters for detecting flooding attacks against SIP based services," *Computers & Security*, vol. 28, Oct. 2009, pp. 578-591.

[6] S. Ehlert, D. Geneiatakis, and T. Magedanz, "Survey of network security systems to counter SIP-based denial-of-service attacks," *Computers & Security*, vol. 29, Mar. 2010, pp. 225-243.

[7] Cisco, "Voice Security Primer: Protecting the Voice Infrastructure."

[8] Juniper Networks, "Enterprise VoIP Security," Apr. 2006.

[9] UK Security Service., "CRAMM: The UK Government's Risk Analysis and Management Method."

[10] K. Brand and H. Boonen, *IT Governance based on Cobit 4.1 - A Management Guide*, Van Haren Publishing, 2007.

[11] M. Swanson, "Generally Accepted Principles and Practices for Securing Information Technology Systems," Sep. 1996.

[12] C. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, vol. 27, 1948, pp. 623-656.

[13] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol, RFC 3261," 2002.

[14] "Internet Protocol, Darpa Internet Program Protocol Specification, RFC 791," Sep. 1981.

[15] J. Poster, "User Datagram Protocol, RFC768," Aug. 1980.

[16] Darpa Internet Program Protocol Specification, "Transmission Control Protocol, RFC 793," Sep. 1981.

[17] P. Mockapetris, "Domain Names - Implementation and Specification, RFC1035," 1987.

[18] O. Hersent, J. Petit, and D. Gurle, *IP Telephony: Deploying Voice-over-IP Protocols*, Wiley, 2005.

[19] M. Arango, A. Dugan, I. Elliott, C. Huitema, and S. Picket, "Media Gateway Control Protocol (MGCP), RFC 2705."

[20] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications, RFC 3550," 2003.

[21] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinet, P. Leach, and T. Berners-Lee, "Hypertext Transfer Protocol HTTP/1.1, RFC 2616," Jun. 1999.

[22] R. Sharda, S. Voß, W.K. Ehrlich, K. Futamura, and D. Liu, "An Entropy Based Method to Detect Spoofed Denial of Service (Dos) Attacks," *Telecommunications Modeling, Policy, and Technology*, S. Raghavan, B. Golden, and E. Wasil, Eds., Springer US, 2008, pp. 101-122.

[23] J. Mirkovic and P. Reiher, "D-WARD: a source-end defense against flooding denial-of-service attacks," *Dependable and Secure Computing, IEEE Transactions on*, vol. 2, 2005, pp. 216-232.

[24] J. Wang, L. Lu, and A.A. Chien, "Tolerating denial-of-service attacks using overlay networks: impact of topology," *Proceedings of the 2003 ACM workshop on Survivable and self-regenerative systems: in association with 10th ACM Conference on Computer and Communications Security*, Fairfax, VA: ACM, 2003, pp. 43-52.

[25] T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of Network-based Defense Mechanisms Countering the DoS and DDoS Problems," *ACM COMP. SURV*, vol. 39, 2007.

[26] G. Gandhi and S. Srivatsa, "An Entropy Algorithm to Improve the Performance and Protection from Denial-of-Service Attacks in NIDS," *Computer and Electrical Engineering, 2009. ICCEE '09. Second International Conference on*, 2009, pp. 603-606.

[27] J.C. Pelaez and E.B. Fernandez, "VoIP Network Forensic Patterns," *Proceedings of the 2009 Fourth International Multi-Conference on Computing in the Global Information Technology*, IEEE Computer Society, 2009, pp. 175-180.