User identification in encrypted network communications

Koch, R.; Rodosek, G.D.

1. This is the camera ready version of the paper "User identification in encrypted network communications," © IEEE.

2. Please use the citation of the original publication:

doi: 10.1109/CNSM.2010.5691292
keywords: {cryptography; statistical analysis; telecommunication traffic; encrypted network communication; encrypted traffic; keystroke dynamics; monitored network packet; network environment; network traffic clustering; network traffic encryption; statistical analysis; statistical evaluation; user identification; user profiles; Correlation; Cryptography; Delay; Intrusion detection; Protocols; Servers},
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5691292&isnumber=5691186

3. Please find the article abstract in IEEE Xplore:
Abstract—Encrypting network traffic is a normal procedure to protect information for exchange. This prevents tapping and manipulation but it also hampers intrusion as well as data leakage and misuse detection. Obtaining knowledge about users of encrypted communications is, however, beneficial in terms of monitoring access, security and accounting reasons. Thus, the objective is to provide evidence of the source of actions, especially to detect insiders and illegal connections, without the necessity of decrypting the network traffic.

We propose a novel architecture to identify users of encrypted traffic in a network environment of a company. It is based on statistical evaluation of monitored network packets. The proposed approach utilizes and combines two main aspects, the mode of operation of remote sessions and the keystroke dynamics of users. Aspects such as capturing and clustering network traffic, generating user profiles and patterns, and statistical analysis are part of the architecture.

I. INTRODUCTION

One of the most challenging endangerments today is the insider threat. With the knowledge of security mechanisms in place, it is easy for an insider to overcome them, especially with the availability of accounts or access data. Even if the released numbers differ substantially, security studies number the proportion of insiders from at least 20 up to 70% and higher of all security incidents (e.g., see [1] and [2]).

Therefore, approaches must be developed to detect insiders by taking into account that the traffic is encrypted. A decryption for an analysis is undesirable or impossible because of the processing power for the de-/enciphering, unwanted modifications of the infrastructure or protocol or legal restraints. The particular threat extinguishing from an insider is based on his access rights and the knowledge of the infrastructure. So, even if there are strong security mechanisms in place (e.g. a separated network for classified information), to act against an insider is a challenge. If encrypted channels can be used for communication with systems outside of the company, data leakage is hardly detectable even if there is an Intrusion Detection System (IDS) in place. Moreover, encryption is used by more and more services (e.g. with SSL), in addition to the ciphering of the company’s network. Therefore, procedures like deep packet inspection are not applicable in those environments.

The knowledge of the user of an encrypted channel can be important for various reasons such as behavior analysis, misuse detection and accounting. The user profile can be used to build a behavior model of the system. This allows anomaly detection in user behavior: A model of the user and network behavior is built and later on, measurements of the real environment are compared to the expected values of the model. If the difference is beyond a defined threshold, an alarm is raised. Based on that, a protection against data leakage can be built in secure environments without compromising the system by data decryption in security devices.

Our scenario is defined as follows: A company is using encryption for the remote sessions with their servers. The employees are able to initiate encrypted channels to the outside.

Another field of application is accounting. Here, the knowledge of the user and the amount of data transferred can be used for charging purposes or for implementing quality of service mechanisms without analyzing the intrinsic traffic.

The paper is organized as follows: In Section II, a brief overview of related work done in the area of IDSs for encrypted networks is given and secondly the latest work for the required biometrical techniques. Section III introduces a new architecture for an Extrusion Detection System (EDS) capable of detecting insider threats in encrypted networks. The prototypical implementation is described in Section IV and first results are discussed. Finally, Section V contains concluding remarks and describes the next research steps.

II. RELATED WORK

There are currently three basic approaches to carry out intrusion detection on encrypted communication, namely:

- Protocol-based: Detection of misuse of the encryption protocol
- Intrusive: Modifications of the network infrastructure or the encryption protocol
- Non-Intrusive: Statistical analysis of encrypted traffic

ProtoMon is a system developed by Joglekar et al. [3] which instruments shared libraries for cryptographic and application level protocols for conducting intrusion detection. Monitoring is integrated into the protocol handling. By that, attacks on the encryption protocol can be detected. Nevertheless, malicious activities hidden inside the encrypted channel could not be detected.

Intrusive techniques are used by Goh et al. They proposed an IDS for encrypted networks which is able to analyze the
payload and simultaneously maintaining the confidentiality of the encrypted traffic [4]. The network traffic of a sender is replicated and sent to the receiver and also to the Central IDS (CIDS) which is a separate and dedicated host. The protocol is set onto an underlying VPN (guaranteeing the confidentiality) and adds an additional layer to the network layer. The system is able to do payload analysis and to keep the confidentiality, but it strongly depends on modifications of the protocols and infrastructure. It can supervise the VPN of a company, but additional communications protected by e.g. SSH or SSL cannot be analyzed. Therefore it is not deployable in general, also the CIDS is a potential target to attacks and a single point of failure.

Foroushani et al. proposed a system based on the evaluation of the transferred packet sizes and the time intervals between messages [5]. Attacks are detected without a decryption by the use of intrusion signatures which are generated from the frequency of accesses and specifications of the TCP traffic. Anyway, because of a high false alarm rate (about 20% in the best case), the system is not usable for a production environment. The system requires behavior profiles for the target servers and the exchanged information, which are often not available.

Other work addressing IDSs in encrypted environments can be found, but to the best of our knowledge, all of it can be assigned to one of the three categories named before (e.g., see [6] or [7]). Thus, all of these systems are not appropriate for our scenario due to the shortcomings shown.

In the area of user authentication, keystroke dynamics has been analyzed. Sharif et al. [8] proposed an authentication system which combines the conventional user/password method with biometric techniques, keystroke dynamics and click patterns. They used the maximum- and minimum time interval between two specific keystrokes, the order and the measurement of delays when typing, the constant between the keystrokes and the click pattern of the user in a colored square with four elements. An error margin was introduced by Sharif which divides the users into three groups (experts, standard, beginners), based on their typing abilities. In [9], Rybnik et al. investigated an authentication system based on keystroke dynamics using a fixed text. They summarized the following seven points for the feature extraction of keystrokes: Dwell (duration of a specific keystroke), flight (pause between keystrokes), typing speed, overlapping of specific key combinations (e.g. by fast typing or using the shift key), amount of errors (detected by the use of delete- or backspace), method of error correction and cursor navigation keystrokes.

Other procedures for biometric user identification can be based on the analysis of the rest-periods during playing a computer game [10].

In [11], Melnikov et al. showed that an user identification is possible by the evaluation of flow packets derived from the browsing behavior of a user. First results demonstrated that by using a cross correlation between the flow traces and the stored profiles, good classification results could be achieved on the user group (due to the early state, only four profiles had been used yet).

III. A NOVEL ARCHITECTURE FOR EXTRUSION DETECTION

To overcome the deficiencies of current approaches regarding encrypted network connections, we present a novel architecture for the identification of insider threats. An overview of the structure of the system architecture is shown in Figure 1.

The system utilizes and combines two main aspects, the (i) mode of operation of remote sessions and the (ii) keystroke dynamics of the users.

A remote session starts with the authentication of the user when connecting the server. After the user has entered the login information, it is transmitted to the server where it is verified and the remote session is built-on. All subsequent user inputs are transferred directly to the server. Because of the usability of the session, every keystroke or mouse movement is transferred immediately. Every keystroke is sent by a network packet with a characteristic payload size (e.g., 48 bytes when using SSH v2 with the default cipher AES128–CBC). The server sends an acknowledge (ACK) (zero-sized payload) and an echo packet of the keystroke. When executing a command by pressing Enter, the server sends the result with one or more network packets, which are typically much bigger than the keystroke packets. With the reception of the server response, a command is completed and no packets are exchanged until the next keypress (except possible keep-alives, etc.). Therefore, the keystroke dynamics of the user can be retrieved by the timing and size of the observed encrypted packets.

We define a cluster as followed: A cluster contains all transferred packets of a completed command, composed of the entered command itself and the corresponding answer from the server, but excluding acknowledge- and echo-packets.

Consequently, the first step for the statistical evaluation of the transmitted packets is the allocation of the consecutive clusters.

Not all keystroke features are available in an encrypted session. The following parameters can be used: Flight (time difference between two encrypted packets), typing speed, maximum- and minimum interval between two specific keystrokes, regularity (average of inter-arrival-times), order of delays (e.g., the delay between the second and third key pressed can be typically longer than the average delay between the fourth and fifth key pressed by a specific user), delay values (time between two following keystrokes).

Additional, a parameter called profile delay is calculated as followed: When the user keystrokes are identified after the completion of a cluster, After a complete cluster has been detected, all delays between every two successive keystrokes are determined and ordered afterwards. For example, five keystrokes of the input mount are captured with the following delays (for clarity, only entire seconds are used):

\[ m \ (1) \ o \ (7) \ u \ (4) \ n \ (3) \ t \ (8) \ Enter \]
Then the profile delay for the typing sequence of the user would be, based on the ascending delays:

\[1 - 4 - 3 - 2 - 5\]

The results of the feature evaluation (cross correlations and absolute differences) are summed up for every user profile. The user profile with the highest values is assigned to the cluster. For a better classification result, the evaluation of a series of clusters can be summed up.

IV. PROOF OF CONCEPT AND FIRST RESULTS

For the further evaluation and development of the concept, a first prototype of the proposed architecture was implemented. For the statistical analysis, an user profile is generated for every employee. The profile consists of the parameters available in encrypted sessions as presented in Section III. Therefore, each user has to enter multiple sentences and commands whereof the clusters and parameters are obtained just like in the operational analysis of the network data, as shown subsequent. The extracted keystroke features and the cluster data are saved into a database.

The network traffic is wiretapped by the use of the libpcap-library. pcap_compile() and pcap_setfilter() can be used to compile and enable filter expression according to the tcpdump-definition. This enables an easy filtering of all wanted SSH-traffic while the unencrypted traffic is omitted. For every selected network packet, the Unix time stamp, the running number, the size, the source address and the time difference \(\Delta t\) since the last observed packet are recorded. These values are the data for each packet pertaining to a cluster. The accuracy of the time stamp, gained by the function \(\text{gettimeofday}()\), is about one microsecond, which is precise enough for the network environment as well as the required accuracy for the keystroke dynamics. For the functionality of the system, the correct clustering of the network traffic is of key importance. A cluster is detected when the following requirements are fulfilled/detected: (1) A series of keystroke-sized packets sent by the client, (2) the corresponding sequence of ACK/echo packets, (3) an answer of the server exceeding the defined payload threshold, (4) the beginning of a new cluster by the first new keystroke of the client after the defined minimum payload threshold and time (cluster break time). Every finished cluster is passed to the statistical analysis where multiple calculations are done. First, cross correlations between the extracted cluster data and the ones recorded in the user profiles are calculated and saved as probability-counters \((pc)\) for each profile. After that, the keystroke features are compared to the typical keystroke features represented by the user profiles. The \(pc\) of the profile which has the lowest difference to the examined cluster is increased by one. The last step is calculating cross correlations between the keystroke features of the cluster and those of the user profiles and also adding them to the \(pc\). By that, the evaluation of a cluster is finished and the cluster is assigned to the user with the highest value of the \(pc\). Figure 2 shows the adding up of the results of the subsequent analyses. Easy to see, that profile four has a much higher \(pc\) and therefore user four is assigned to the evaluated cluster.

For the first experiments with our prototype, profiles of four users had been created and added into the database. Afterwards, the encrypted network packets of 16 sessions each consisting of five user inputs had been recorded and evaluated by our prototype. Table I shows the classification results. Even only a small number of users had been involved in the first test, an in-depth analysis of the evaluation has shown that the users can be clearly differentiated by our selected features. The scalability of our system under high user numbers will be investigated as a next step. The evaluation was done once by only using the cluster features and then repeated by using the cluster features and the keystroke features as described above. Easy to see, when including the keystroke features for the analysis, the results are clearly more accurate. The row \textit{No Cluster Verification} presents the values of the automated run of the prototype while the row \textit{With Cluster Verification} shows the results, when the alignment of the clusters was

\begin{table}[h]
\begin{center}
\begin{tabular}{|l|c|c|}
\hline
Used Features & Clusters & Clusters and Keystrokes \\
\hline
No Cluster Verification & 50\% & 87,5\% \\
With Cluster Verification & 75\% & 93,75\% \\
\hline
\end{tabular}
\end{center}
\caption{Classification results of the prototype. The evaluation was done based on 4 user profiles and the recording of the network packets of 16 sessions consisting of about 5 user inputs and the corresponding server responses.}
\end{table}
verified. As one can see, if all clusters are identified correct, the results are much better. Therefore, the correct detection of the single clusters is of key importance for our architecture. The evaluation has shown, that especially when a user is interacting with the server very fast or uses a high number of tabulator or delete keystrokes, the system is not able to recognize all clusters appropriately yet.

V. CONCLUSION AND FURTHER WORK

In the paper we proposed a novel architecture for an extrusion detection system to identify users in encrypted networks. First results of our Proof of Concept showed that an user identification is possible without the necessity of a decryption of the network traffic. The knowledge of the user of an encrypted session can be important in different scenarios, for example for intrusion detection, data leakage detection or accounting.

Our proposed architecture utilizes the mode of operation of remote sessions and the keystroke behavior of the users. The network traffic clustering has a key role for the whole system because therein the keystroke sequences of the users are reconstructed. Therefore, the performance of the whole system strongly depends on the quality of the clustering process. As we have seen in our experiments, this process is not yet able to identify all clusters properly. To improve the evaluation, the next step is to analyze and to use dependencies between clusters. Therefore, the probability of different cluster sequences will be evaluated and used to detect allocation mistakes and improve the cluster borders. Also, additional techniques like the analysis of rest-periods and user pause times can be used to improve the accuracy of the system.

An extension of the system is to cross check user actions with different security rules defined in the database. For example, if a connection to a server is activated and the analysis results in a classification Confidential of the session, the connection can be interrupted by the system if a transmission over a defined threshold is made. We are going to realize this by the development of a module for the analysis of the user behavior in our architecture.

ACKNOWLEDGMENT

This work is done at the Chair for Communication Systems and Internet Services led by Prof. Dr. Dreo Rodosek, part of the Munich Network Management (MNM) Team.

REFERENCES