An Efficient Practice of Privacy Implementation: Kerberos and Markov Chain to Secure File Transfer Sessions

Fadi Al-Ayed, Chunqiang Hu and Hang Liu
(Corresponding author: Fadi Al-Ayed)

School of Electrical Engineering and Computer Science, The Catholic University of America
Washington, DC 20064, USA
(Email: 93alayed@cua.edu)

(Received May 1, 2017; revised and accepted July 25, 2017)

Abstract

Kerberos is a widely used authentication and authorization protocol that allows a client to communicate with servers to authenticate mutually, and obtain authorization and tickets to access application services securely with encryption over a non-secure network. It is designed to provide strong confidentiality, integrity, and authentication through symmetric key cryptography. However, Kerberos system itself needs to be monitored and protected. In this paper, we present the implementation of an intrusion detection system (IDS) that monitors a Kerberos network, and reports and responds to malicious activities or policy violations. The designed IDS system performs anomaly activity detection using a Markov fingerprinting scheme that builds a Markov chain model of the normal Kerberos session messages and applies the machine learning technique to detect deviations from the model of normal traffic. We mainly focus on detecting and averting intrusion attempts on File Transfer Protocol (FTP) applications for corporate systems. The proposed scheme can be extended to support other applications on a Kerberos network. The results of our experiments show that implementing Markov fingerprinting with Kerberos can improve the security in terms of prevention and detection of malicious behaviors.

Keywords: Encrypted Traffic; File Transfer Protocol; Kerberos; Markov Model

1 Introduction

Secure data transfer sessions in corporate networks using File Transfer Protocol (FTP) is becoming a vital process. The traditional authentication scheme that simply verifies a user through a login name and password transmitted across the network with plain text is vulnerable in terms of security and privacy because information sent across the network may be intercepted and subsequently used to impersonate the user maliciously. In addition, different services are provided to multiple users on the corporate networks, which requires the ability to identify the authorization of the user to access different services [8,16,17]. Kerberos is a network authentication and authorization protocol that allows clients and servers to authenticate each other, and a client to obtain authorization and tickets to access application services securely with encryption over a non-secure network. It is designed to provide strong confidentiality, integrity, and authentication through symmetric key cryptography [11,15,30]. The main idea under Kerberos is to send a hash of the user's password over the network instead of the actual password itself. The password is checked on both sides of the connection for authentication and authorization, and is also used as a key for symmetric hash encryption. A timestamp is used with the assumption that clients have loosely synchronized time. Kerberos enables a client to be authenticated and authorized to access multiple servers spontaneously. It also has very strong protection against eavesdropping and replay attacks.

Kerberos has been widely used. For example, Windows networking uses Kerberos as its preferred authentication method, with which a client joins a Windows domain, proves its identity for authentication, and obtains the authorized services in the domain and all domains with trust relationships to that domain.

Note that Secure Sockets Layer (SSL), or Transport Layer Security (TLS) is another known network security protocol that provides privacy and data integrity between two communicating computer applications [3]. SSL is a standard security technology for authentication and establishing a secure link between the web server and browser. However, we consider enterprise applications in this paper for which Kerberos provides a simple and general framework for securing the network applications.

Kerberos system itself needs to be monitored and protected. In this paper, we present an implementation of an intrusion detection system (IDS) that monitors a Kerberos network, and reports and responds to malicious activities or policy violations. The designed IDS system detects anomaly activities using a Markov chain to build a stochastic model to represent Kerberos session states. Markov chains [2,9] have been utilized in information processing by taking an advantage of its ability to capture statistical regularities in the behavior of systems and allowing state estimation and pattern recognition. The session states reflects the Kerberos protocol and messages in single-directional traffic flows from client to server or from server to client for an application. Based on the Markov chain model of a normal Kerberos session for a particular application, we perform anomaly detection by detecting deviations from the model of normal traffic using a learning technique, which forms Markov fingerprinting. We seek to implement the state estimation and pattern recognition provided by the Markov chain model to come up with a more pro-active approach for the detection and avoidance of malicious behaviors in Kerberos protocol communications. By establishing a fingerprint through identifying the possible sequence of messages, we would be able to detect abnormal behaviors that might be attempts of intrusion. We mainly focus on detecting and averting intrusion attempts on the FTP type of applications as it is popular for corporate systems. The proposed scheme can be extended to support other applications on a Kerberos network.

The remainder of this paper is organized as follows: Section 2 discusses the related work. Section 3, presents the design methodology. Experimental results are given in Section 4, and Section 5 concludes the paper.

2 Related Work

Computing and software development have evolved from the noble objectives of providing solutions for daily problems to more malicious intents. Disrupting and stealing, or harming, have been obfuscations items for quite a while and calls for stricter implementations on data security. Computer systems are still human driven, and as such, there is an element of unpredictability. This then led to quite a growing effort to infuse some pattern and behavioral recognition capabilities in the applications and the protocols they are running on. In the study of intrusion detection, and probabilistic techniques have been used which would be represented as decision trees.

Ye *et al.* made several tests for Markov chain which was applied to set of computer audit data to investigate the frequency and ordering property of the information [28]. The study gave answers to questions on which properties were critical to detect intrusions techniques that were based on the frequency property provided solution that resulted to good intrusion detection.

Abraham *et al.* conducted a study of implementing an attack graph that was very similar in characteristics to a decision tree [2]. With the combination of vulnerabilities observed in a network configuration, several scenarios were built where an attacker can reach the goal state [10]. Based on the attack graph, Markov chain simulation was conducted.

Other studies focused on the encrypted network traffic flow and their classifications. Korczynski *et al.*, on traffic encryption, postulated a method that uses the size of first few packets of data as a basis to enable early application recognition [2,12,19,23]. Another method tried identification of SSL/TLS encrypted application layer protocols using a signature-based and a flow-based statistical analysis process.

Signature recognition technique is the term that Ye *et al.* used for the ability to identify anomalies in behavior and signal intrusions. He developed an anomaly detection process that corresponds to the norm profile of a temporal behavior via Markov model [5,27]. Qassim *et al.* [21] proposed a network anomalies classifier which utilizes machine learning to classify activities detected and to monitor network behavior by a packet header based anomaly detection system.

The aforementioned works have all been centered on the utilization of the Markov chain to be able to come up with signatures of behavior that can be used to identify intent. Our work, however, is an advance approach in this direction to enhance the traditional way which would be revolving around the same concept.

3 Methodology

In this section, we describe the implementation of the Markov chain based intrusion detection scheme for Kerberos sessions.

3.1 Kerberos Overview

As shown in Figure 1, a Kerberos system consists of Authentication Server (AS), Ticket Granting-Ticket (TGT), and application server [24]. A client sends a Ticket-Granting-Ticket (TGT) request to the AS, the AS verifies the access right in database based on the user's ID, generates a TGT and session key, and send them to the client in a reply message. The TGT is a token which enables the user to request access to application services without re-supplying its credentials. The TGT is encrypted by a TGS key that TGS knows, but the user would not know. The session key is encrypted using a key derived from the user's password. The client will ask for the user's password and use it to decrypt the incoming reply message to get the client-TGS session key after it receives the reply from the AS. When the client first attempts to access an application service, it sends the request message for the application service granting ticket to the TGS. This request message contains TGT and an authenticator that contains user ID, network address, and timestamp, and is encrypted by the client-TGS session key.

The TGS decrypts the TGT and authenticator, verifies the request, generates a granting ticket for the particular requested application server if it is authorized, and a client-application server session key, and sends the reply to the client. The ticket is encrypted by an application server key that knows by the application server but not the client, and the client-application server session key is encrypted with the client-TGS session key. After the client receives the reply message from TGS, it obtains the application service granting ticket. To access the application service, the client sends the request message to the application server, which contains the application service granting ticket and authenticator. The authenticator is encrypted with the client-application server session key. The application server verifies the ticket and authenticator, and grants access to the application service according to the authorization data specified in the ticket. The connection between the user and the application service is thus established. If mutual authentication is required, the server returns an authenticator to the client. Table 1 summarizes the Kerberos messages exchanges, and Table 2 presents the notation of the negotiation parameters between the server and client [24, 26].

Table 1: Summary of Kerberos message exchanges

(1)	**C → AS**	$Options \parallel ID_c \parallel Realm_c \parallel ID_{tgs} \parallel Times \parallel Nonce_1$
(2)	**AS → C**	$Realm_c \parallel ID_C \parallel Ticket_{tgs} \parallel \mathrm{E}(K_c, \parallel K_{c,tgs} \parallel Times \parallel Nonce_1 \parallel Realm_{tgs} \parallel ID_{tgs}])$
		$Ticket_{tgs} = \mathrm{E}(K_{tgs}, [Flags \parallel K_{c,tgs} \parallel Realm_c \parallel ID_C \parallel AD_C \parallel Times])$

(a)Authentication Service exchange to obtain ticket-granting ticket

(3)	**C → TGS**	$Options \parallel ID_v \parallel Times \parallel Nonce_2 \parallel Ticket_{tgs} \parallel Authenticator_c$
(4)	**TGS → C**	$Realm_c \parallel ID_C \parallel Ticket_v \parallel \mathrm{E}(K_{c,tgs}, [K_{c,v} \parallel Times \parallel Nonce_2 \parallel Realm_v \parallel ID_v])$
		$Ticket_{tgs} = \mathrm{E}(K_{tgs}, [Flags \parallel K_{c,tgs} \parallel Realm_c \parallel ID_C \parallel AD_C \parallel Times])$
		$Ticket_v = \mathrm{E}(K_v, [Flags \parallel K_{c,v} \parallel Realm_c \parallel ID_C \parallel AD_C \parallel Times])$
		$Authenticator_c = \mathrm{E}(K_{c,tgs,} [\ ID_C \parallel Realm_c \parallel TS_1])$

(b)Ticket-Granting Service exchange to obtain service-granting ticket

(5)	**C → V**	$Options \parallel Ticket_v \parallel Authenticator_c$
(6)	**V → C**	$\mathrm{E}_{K,}\mathrm{c,v}\ [TS_2 \parallel Subkey \parallel Seq\#]$
		$\mathrm{Ticket}_v = \mathrm{E}(K_v, [\mathrm{Flags} \parallel K_{c,v} \parallel Realm_c \parallel ID_C \parallel AD_C \parallel Time_s])$
		$Authenticator_c = \mathrm{E}(K_{c,v}, [ID_C \parallel Realm_c \parallel \mathrm{TS}_2 \parallel Subkey \parallel Seq\#])$

(c) Client/Server Authentication exchange to obtain service

Table 2: Notation

Notation	Description
AS	Authentication Server
C	Client
V	Server
ID_C	Identifier of user on C
ID_V	Identifier of V
P_C	Password of user on C
AD_C	Network address of C
||	String concatenation operation
K_V	Secret encryption key shared by AS and V

The remaining attributes are described as follows [24], *Realm*: Indicates realm of user. *Options*: Used to request that specific flags be sent in the returned ticket. *Times*: Used for time settings in the ticket e.g. desired start time, and expiration time for the requested ticket. *Nonce*: A random value to be repeated in message to assure that the response is fresh and have not been replayed by an opponent. *Subkey*: This field is omitted, since the session key from the ticket (KC,V) is used. *Seq*: This optional field that specifies the starting sequence number to be used by the server for messages sent to the client during this session.

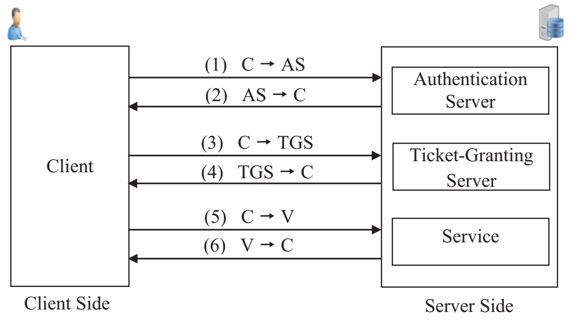

Figure 1: Overview of kerberos

3.2 Markov Model Fingerprint

In this section, we illustrate an approach based on Markov chains to model possible sequences of message types observed in a single-directional Kerberos session.

Consider a discrete-time random variable X_t for any $t = t_0,\ t_1, \ldots, t_n \in T$ such that X_t where $1 \leq t \leq n$. It takes values $i_t \in \{1, \ldots, s\}$ that corresponds to the observed Kerberos message types during a session. Assuming that X_t is the first-order Markov chain [12, 13, 29], and P is denoted as the transition matrix then:

$$P\left(X_t = i_t \mid X_{t-1} = i_{t-1}, X_{t-2} = i_{t-2}, \ldots, X_0 = i_0\right) = P\left(X_t = i_t \mid X_{t-1} = i_{t-1}\right). \tag{1}$$

Assuming further that the Markov chain is homogeneous, i.e. state transition is time-invariant:

$$P\left(X_t = i_t \mid X_{t-1} = i_{t-1}\right) = P\left(X_t = j \mid X_{t-1} = i\right) = p_{i,j} \tag{2}$$

Using the transition matrix:

$$P = \begin{bmatrix} P_{1,1} & P_{1,2} & \cdots & P_{1,s} \\ P_{2,1} & P_{2,2} & \cdots & P_{2,s} \\ \ldots & \ldots & \ddots & \vdots \\ P_{s,1} & P_{s,2} & \cdots & P_{s,s} \end{bmatrix}$$

Where:

$$\sum_{j=1}^{s} P_{i,j} = 1.$$

The function:

$$Q = [q_1, q_2, \cdots, q_s], \tag{3}$$

is denoted as the Initial Probability Distribution (IPD) where q_i=P(X_t=i) at time t=0, and the following function:

$$W = [w_1, w_2, \cdots, w_s], \tag{4}$$

is denoted as the Exit Probability Distribution (EPD) where wi characterizes the probability that the Kerberos session finishes when if it is in state i at time tn. However, both initial and exit probability distributions are independent of the Markov chain.

Lastly, the probability that a sequence of states $X_1, \ldots, X_T$ corresponding to a single Kerberos session occurs is denoted by:

$$P(X_1, \ldots, X_T) = q_i \prod_{t=2}^{T} P_{i_{t-1}}, i_t \times W_{iT}. \tag{5}$$

The observed transition probability matrices and the IPD for the Kerberos sessions can be used by the Markov Classifier to classify traffics and identify traffic anomaly as described below. For example, an attacker may steal or forge a Kerberos TGT and attempts to gain the application services with is called Golden Ticket attack [8]. Such a Golden Ticket attack can be detected based on the Kerberos traffic anomaly it creates, an attacker sends a valid TGT to the TGS with no prior successful AS requests to obtain a TGT.

3.2.1 Markov Classifier (MC)

The Markov Classifier (MC) uses a first-order homogeneous Markov chain to build a stochastic model that reflects the Kerberos session states. A Kerberos session model is obtained per flow direction that corresponds to each transaction or process used. The decimal codes shown in Table 3 are used to represent the states during a Kerberos session.

Table 3: Kerberos states codes

SID	Description
20:	Username Exist ($Username\_Exits$)
21:	Check Active Director (AD_Check)
22:	Request for TGT ($Ticket\_request$)
23:	Generation of TGT and Session Key ($TGT\_SessionKey$)
24:	Request for Service Ticket ($GS\_Success$)
25:	Generation of Service Ticket and Another Session Key ($e\_h\_Session\_Key$)
26:	Present Service Ticket to Server (Tkt_http)
27:	Authenticate and Approve the use of Service ($Kerberos\_Success$)
28:	Authenticated Users ($Kerberos\_Success = 1$)
29:	Check for Attackers ($Alert\_Exit$)
30:	Unknown Users ($Username\_Exits = 0$)

As shown above, the state of check username exist decimal code is 20: and decimal code 21: is for checking the active directory of the clients (AD_Check). The initial user authentication request ($Ticket\_request$) is represented by a decimal code of 22: The reply of the ($TGT\_SessionKey$) is denoted by 23: which contains the TGT and the client-TGS session key. Decimal code 24: is the request to the TGS ($GS\_Success$) by the client for an application service ticket. The TGS reply ($e\_h\_Session\_Key$) to the request is denoted by 25: which contains the application service ticket and application service session key created by TGS. Decimal code 26: is the client's request to the application server (Tkt_http), which includes the ticket and client authenticator to access the application service. The application server that authenticates the user which is represented by the decimal code 27: The process may also respond ($Kerberos\_Success$) to perform mutual authentication and indicate completion of the process with a code 28: In addition, the alert state decimal code is 29: and the state of unknown users decimal code is 30.

3.2.2 Learning and Ranking Mechanisms

We utilize the rule-based learning algorithm to classify the Kerberos session Markov state parameters for different types of users [4, 18, 31]. Table 4 shows the variables.

Table 4: Notation and description of each variable

Notation	Description
$q_{(n)}$	Represents each Kerberos parameter
q	Sum of indexed Kerberos values
$x_{(m)}{}^{(n)}$	x=values in session, n=counter, and m=user type
$y^{(n)}$	Sum of previous session
?	Queries of summed sessions
h_x	Results of desired queries

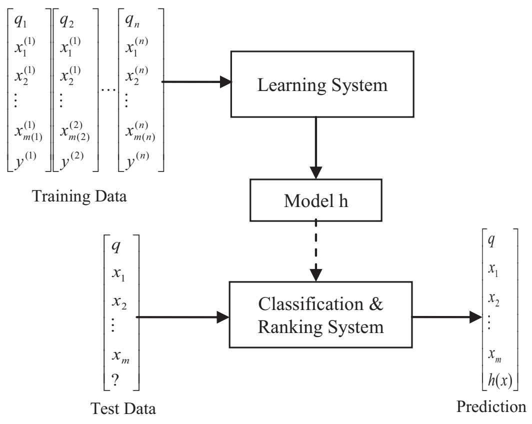

Figure 2: Classification architecture

Figure 2 shows the learning and classification architecture. To build the Markov models of the Kerberos session behaviors for different type of users, the Markov classifier is first trained. We use the Kerberos session data for normal users and known attacks to train the MC and build the Markov models. The messages of Kerberos sessions are collected and pre-processed. Essential features that can differentiate one session from the other such as user IDs, IP addresses and ports are identified. The MC analyzes the messages, calculates the Markov model parameters for each type of users, and builds the models for normal authorized users and known attacks as discussed later in this section. The trained classifier is used to detect malicious activities on the test data. The data captured in the test phase is classified based on the Markov chain state transition fingerprints of different types of users, normal users and adversaries which will have different parameters of the Markov models. The decision process for session classification and anomaly detection is based on both the likelihood criterion and the likelihood ratio criterion as described in the next subsection.

Figure 3 illustrates an overview of the MC-based classifier operations. It consists of data flow which records every single activities messages exchanged during Kerberos sessions into a log and control flow to estimate the execution of the following operations: likelihood criterion and the likelihood ratio criterion.

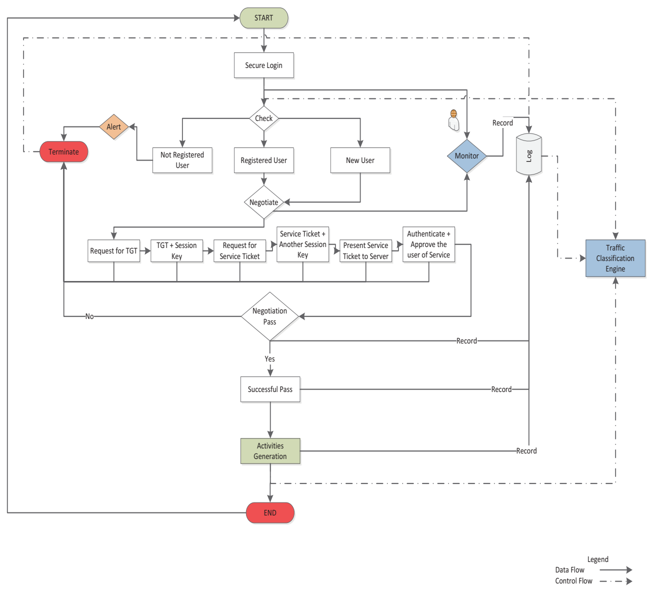

Figure 3: Overview of operational architecture

To begin with data flow (new session), a user is required to enter the granted username and password on the login page. The client has three attempts to login. If the user fails the attempts, the account will automatically be locked out and a notification will be sent to both original client and system administrator. If the client has been successfully logged in, the system checks for the previous user's information recorded in the database for further security purpose. Each client should have one secure login page, in case the client changes the workstation, another secure login page must be generated by the system administrator, otherwise it begins a Kerberos negotiation process. For each user attempt: first, second or third, a new Kerberos session is created. Hence, a new Kerberos ID is related to each attempt. The negotiation process is as flows: the client requests for TGT, if there is no response from either the client or the server, the data flow connection will be terminated and the data of this activity is recorded into a log. The mechanism process applies the same for other Kerberos parameters. More precisely, every single activity in data flow is recorded into a log either if its succeed or failed. The determination of data flow is maintained by control flow for further execution process to classify a session. The execution of control flow will be studied in the next section.

Algorithm 1 shows how the MC classifier computes the transition matrix based on the messages exchanged during Kerberos sessions.

Algorithm 1 Working of Markov

```
1:  Begin
2:  Choose Initial parameter estimates
        $$Q = [q_1, q_2, \cdots, q_s],$$
3:  **if** Username Exit is invalid **then**
4:      Drop the message
5:  **else**
6:      Compute:
        $$P(X_t = i_t \mid X_{t-1} = i_{t-1}, X_{t-2} = i_{t-2}, \ldots, X_0 = i_0) =$$
        $$P(X_t = i_t \mid X_{t-1} = i_{t-1}).$$
7:      Compute:
        $$P(X_1, \ldots, X_T) = q_i \prod_{t=2}^{T} P_{i_{t-1}}, i_t \times W_{iT}.$$
8:      Verify the completed negotiated messages:
9:      **if** negotiated messages = True **then**
10:         Accept the user session
11:     **else**
12:         Drop the message
13:         Compute:
            $$W = [w_1, w_2, \cdots, w_s],$$
14:     **end if**
15: **end if**
16: End
```

3.2.3 Session Classification and Anomaly Detection

In order to classify a session and detect potential attacks, the trained MC classifier will process the network traffic and extract the message exchange for a Kerberos session based on the user ID, IP address and port. It analyzes the messages exchanged for a session and makes a classification decision. The message exchanges behavior and Markov model parameters may be known for some kinds of attacks, but unknown for new types of attacks. Thus, the decision process is based on two test criteria [6,7,14,25]:

1) The likelihood to be a normal authorized session;

2) The likelihood ratio of being a normal session to being abnormal session. If one tests fails, it is assumed that the session is potentially an abnormal session.

For likelihood test, the null and alternative hypotheses are:

H_0: A session is normal (Null Hypothesis);

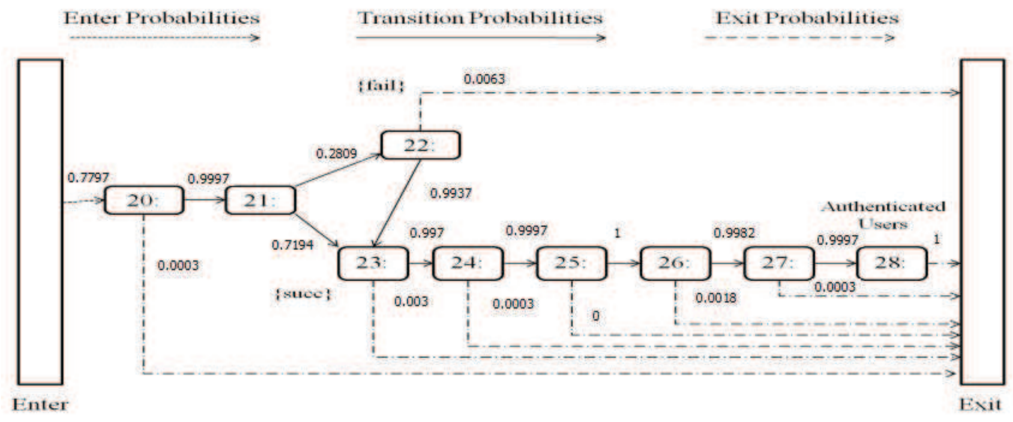

Figure 4: Parameters of the fingerprint for normal users

H_1**:** A session is abnormal (Alternative Hypothesis).

Given the input data sequence corresponding to the observed Markov states during a session, $\{x_1, \ldots, x_T\}$, the likelihood function of being a normal session is equal to:

$$L(H_0|x_1, x_2, \ldots, x_T) = P(x_1, x_2, \ldots, x_T|H_0) \quad (6)$$

The decision rules are if $L(H_0|x_1, x_2, \ldots, x_T) \geq \Delta$, don't reject H_0; if $L(H_0|x_1, x_2, \ldots, x_T) < \Delta$, reject H_0. We can select an appropriate value of likelihood criterion Δ. The impact of the likelihood criterion Δ will be studied in the next session.

For likelihood ratio test, we consider the null and alternative hypotheses which are:

H_0**:** A session is normal (Null Hypothesis);

H_1**:** A session is one of the known attacks (Alternative Hypothesis).

Assume that Markov model parameters of the Kerberos sessions for a set of attack types, $\{\Omega_1, \Omega_2, \ldots, \Omega_k\}$ have been known. Given the observed Markov states during a session, $\{x_1, \ldots, x_T\}$, we can find the maximum value of the likelihood function of being one of the known attack types that is:

$$L(H_1|x_1, x_2, \ldots, x_T) = \arg_{\Omega_i} \max L(\Omega_i|x_1, \ldots, x_T) \quad (7)$$

The likelihood being a type of attacks is the probability of the Markov state sequence computed over such a type of attacks, $L(\Omega_i|x_1, \ldots, x_T) = P(x_i, \ldots, x_T|\Omega_i)$. Then, the likelihood ratio is:

$$\Lambda = \frac{L(H_0|x_1, x_2, \ldots, x_T)}{L(H_1|x_1, x_2, \ldots, x_T)} = \frac{L(H_0|x_1, x_2, \ldots, x_T)}{\arg_{\Omega_i} \max L(\Omega_i|x_1, \ldots, x_T)}. \quad (8)$$

The likelihood ratio test provides the decision rule as follows:

$$\begin{cases} \text{If } \Lambda \geq \Gamma, & \text{do not reject } H_0 \\ \text{If } \Lambda < \Gamma, & \text{reject } H_0 \end{cases}$$

The values Γ is chosen to obtain a specified significance level, and its impact will be studied in the next section.

Finally, we consider H_0 is rejected if either the above likelihood test or the likelihood ratio test fails.

4 Experimental Results

In this section, we show the experimental results and evaluate the performance of the Markov chain model based on anomaly detection scheme.

4.1 Examples of Markov Fingerprints

We have conducted the experiments to obtain Markov chain models for two different types: normal users and attackers. We have also examined the scheme to detect a special type of attack which is known as Golden Ticket attack.

Figure 4 demonstrates the observed parameters of the Markov chain model for normal users. The change of state from decimal code 20: which is check *Username_Exits* to decimal code 21: which is *AD_Check* is probable by 99.97% of sessions, whereas 0.03% are sessions representing failure and closing prior to authentication process. A TGT is set to be valid for 10 hours in our experiments. Hence, this would prevent the client from having any further login issues, instead of going through the entire process of requesting for a new ticket. It is managed through the Active Directory which keeps active record of the clients who logged in within the 10 hours. Whenever a user logs in, the Active Directory (AD) is checked for record, if a TGT exists, the user is allowed to proceed to the server; otherwise, the user has to request for a TGT. From the above Figure 4, 28.1% out of 99.97% requires a new request for TGT and 71.9% out of 99.97% does not. However, at state decimal code 22: which is *Ticket_request* has 99.37% probability to continue to the next state of decimal code 23: which is *TGT_SessionKey*. Furthermore, the flow from the state of decimal code 23: to the state of decimal code 24: which is *GS_Success* is probable by 99.7% of sessions. Consequently, successful authentication state which is decimal code 27: has an 99.97% probability to continue to the next state of decimal code 28: which includes all the successful Kerberos sessions. We observe that the success rate at each state for the normal users is considerably high.

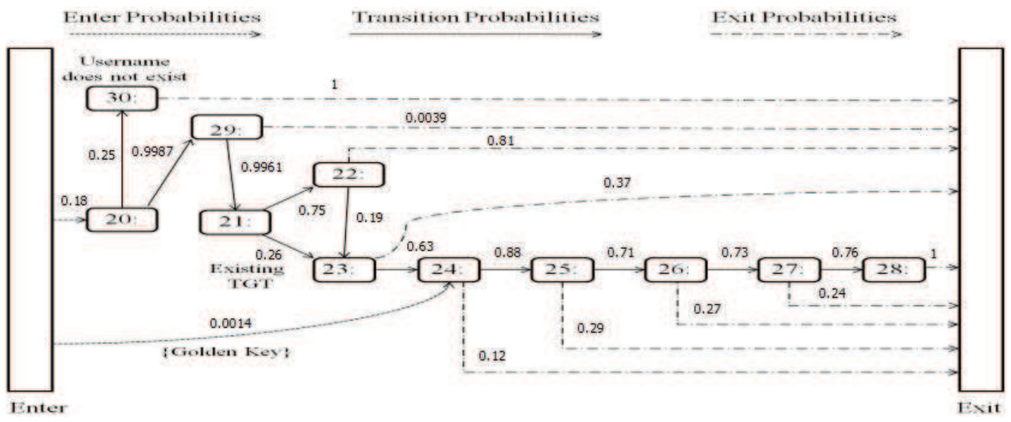

Figure 5: Parameters of the fingerprint for attackers

Figure 5 demonstrates the observed parameters of the Markov chain model for abnormal users (attackers).

As shown above in Figure 5, the change of state from decimal code 20: which is check *Username_Exits* to decimal code 29: which is *Alert_Exit* is probable by 99.87%. At the Alert state, past record of the user's session is checked including the number of attempts compared with the previous successful attempts. Meanwhile, the change of state from decimal code 20: to decimal code 30: which is for unknown users, is probable by 25% which no record exists. We observe that the success rate at each state for the attackers is much lesser.

Golden Ticket Attack, is a special type of attack particularly on the Kerberos approach. In Kerberos, every TGT is valid for certain period of time (10 hours in our case). A golden ticket attacker with get hold of this TGT, has the ability to edit it and reuse it. Typically, the attacker either increases the validity of the ticket or increases its privileges or uses TGTs of deleted users. Since privileges are not used in our case, we have considered the validity of the ticket and TGTs of deleted users. These types of golden attacks are considered on the following criteria:

- Attempts of validity is increased of existing user.
- Attempts of validity is increased of deleted user.

Basically, the attackers edit the TGTs. So there is no original Ticket request associated with the TGTs. To detect the attacks, we examine a Ticket request associated with every TGT, if it is not available, corresponding session which is probable by 00.14% of golden ticket attacks.

This demonstrates the observed parameters of the Markov chain model for golden ticket attacks, with which the adversaries have forged Kerberos TGTs. The adversaries can control every aspect of the forged ticket including the Ticket's user identity, permissions and ticket life time. Since the adversaries use the forget TGT in a TGS-REQ message, they do not need to go through the authentication transaction to create the TGT, or the authentication and authorization transaction does not match the TGT [8].

4.2 Performance of MC-based Classification and Anomaly Detection

We have established a dataset, which was used to evaluate our scheme, consists of 4330 records represents the number of flows identified in Kerberos data.

We conducted the experiments in order to evaluate and to assess the performance of the Markov based classification and anomaly detection: the True Positive Rate denoted as TPR and False Positive Rate denoted as FPR, respectively. True Positive occurs as a class of user Kerberos sessions is correctly classified as the given user session. False Positive arises as another class of user sessions is incorrectly classified as the given class of session. More specifically, the True Positive represents the number of actual attacks classified as attacks, False Positive corresponds to the number of actual normal sessions classified as attacks, True Negative represents the number of actual normal sessions classified as normal sessions, and False Negative corresponds to the number of actual attacks classified as normal sessions [4, 12]. The True Positive Rate (TPR) also represents the detection rate, and can be calculated as [1, 20, 22]:

$$TPR = \frac{TP}{TP + FN} \qquad (9)$$

The False Positive Rate (FPR) can be calculated as:

$$FPR = \frac{FP}{FP + TN} \qquad (10)$$

4.3 Classification Results

We report the classification results of MC classification experiments below. Table 5 shows the results for MC on the dataset.

Using Equation (9), the TPR result is 0.9801, and Equation (10) for FPR which is 0.0264. We notice that the TPR is very large with relatively small rate of FPR.

Figure 6 shows the performance results of evaluating the trade-offs between the prospective true positive rate and false positive rate. The higher the area under

Table 5: Comparision standard martix

		Predicted Label	
		Positive	Negative
Actual Class	Positive	True Positive (3352)	False Negative (68)
	Negative	False Positive (24)	True Negative (886)

the curve top left corner, the better performance of the scheme.

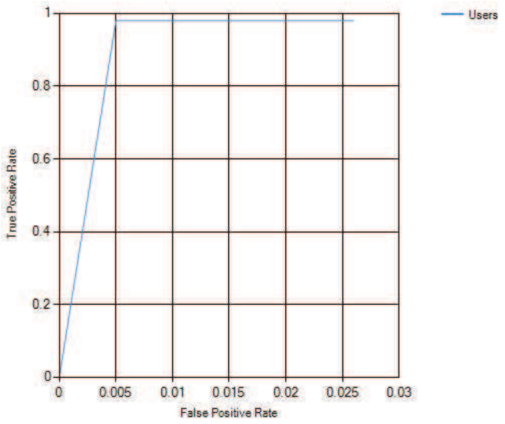

Figure 6: True positive rate vs. False positive rate

A major improvement is noticed with the true positive rate as the curve is very close to the perfect classification point (0,1). Thus, the Markov based classification is very effective.

5 Conclusion

In this paper, we have presented the implementation and evaluation of a Markov chain model based scheme for classification and anomaly detection of Kerberos sessions. The fingerprints of the Kerberos messages are classified based on first-order homogeneous Markov chain and used to detect the anomaly. The evaluation results show that implementing Markov fingerprinting with Kerberos can improve the security in terms of detection of malicious behaviors.

References

[1] A. Abraham, C. Grosan and C. Martin-vide, "Evolutionary design of intrusion detection programs," *International Journal of Network Security*, vol. 4, no. 3, pp. 328–339, 2007.

[2] S. Abraham and S. Nair, "Cyber security analytics: A stochastic model for security quantification using absorbing markov chains," *Journal of Communications*, vol. 9, no. 12, pp. 899–907, 2014.

[3] F. Al-Ayed and H. Liu, "Synopsis of security: Using kerberos method to secure file transfer sessions," in *International Conference on Computational Science and Computational Intelligence*, pp. 1016–1021, 2016.

[4] M. Ambusaidi, X. He, P. Nanda and Z. Tan, "Building an intrusion detection system using a filter-based feature selection algorithm," *IEEE Transactions on Computers*, vol. PP, no. 10, pp. 2986–2998, 2016.

[5] S. Babu, E. Shayesteh and P. Hilber, "Analysing correlated events in power system using fault statistics," in *International Conference on Probabilistic Methods Applied to Power Systems*, pp. 1–6, 2016.

[6] C. N. Barati, S. A. Hosseini, S. Rangan, P. Liu, T. Korakis, S. P. Shivendra and S. R. Theodore, "NTCS: A real time flow-based network traffic classification system," *IEEE Transactions on Wireless Communications*, vol. 14, pp. 6664–6678, 2015.

[7] R. Bardenet, A. Doucet and C. Holmes, "Towards scaling up markov chain monte carlo: An adaptive subsampling approach," in *Proceedings of the 31st International Conference on Machine Learning*, vol. 4, pp. 405–413, 2014.

[8] T. Beery and M. Cherny, *Watching the Watchdog: Protecting Kerberos Authentication with Network Monitoring*, Technical Report, 2015.

[9] Z. Conghua and C. Meiling, "Analysis of fast and secure protocol based on continuous-time markov chain," *Communications, China*, vol. 10, no. 8, pp. 137–149, 2013.

[10] R. Hewett and P. I. Likelihoods, "Exploitability of network vulnerabilities in a large-Scale attack model Kijsanayothin," *International Journal of Network Security*, vol. 17, no. 4, pp. 383–394, 2015.

[11] N. Kamesh and S. Priya, "Security enhancement of authenticated rfid generation," *International Journal of Applied Engineering Research*, vol. 9, no. 22, pp. 5968–5974, 2014.

[12] M. Korczynski, "Classifying application flows and intrusion detection in internet traffic," pp. 1–138, 2012.

[13] M. Korczynski and A. Duda, "Markov chain fingerprinting to classify encrypted traffic," *Proceedings - IEEE INFOCOM*, pp. 781–789, 2014.

[14] S. Liu, A. Takeda, T. Suzuki and K. Fukumizu, "Robust density ratio estimation: Trimming the likelihood ratio," pp. 25, 2017.

[15] A Lowdell and Johan Sakbas, "Achieving authentication and authorization in the combine system," no. 5, pp. 1–25, 2009.

[16] R. Muradore and D. Quaglia, "Energy-efficient intrusion detection and mitigation for networked control systems security," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 3, pp. 830–840, 2015.

[17] T. A. T. Nguyen and T. K. Dang, "Combining fuzzy extractor in biometric-kerberos based authentication protocol," in *Proceedings International Conference on Advanced Computing and Applications*, vol. 23, no. 2, pp. 1–6, 2016.

[18] S. Pan, T. Morris, S. Member, U. Adhikari and S. Member, "Developing a hybrid intrusion detection system using data mining for power systems," *IEEE*

Transactions on Smart Grid, vol. 6, no. 6, pp. 3104–3113, 2015.

[19] S. S. L. Pereira, J. L. C. Silva and J. E. B. Maia, "Ntcs: A real time flow-based network traffic classification system," in *Proceedings of the 10th International Conference on Network and Service Management*, vol. 13, pp. 368–371, 2015.

[20] E. Popoola and A. Adewumi, "Efficient feature selection technique for network intrusion detection system using discrete differential evolution and decision tree," *International Journal of Network Security*, vol. 19, no. 5, pp. 660–669, 2017.

[21] Q. S. Qassim, A. M. Zin and M. J. A. Aziz, "Anomalies classification approach for network-based intrusion detection system," *International Journal of Network Security*, vol. 18, no. 6, pp. 1159–1172, 2016.

[22] K. K. Ravulakollu, "A hybrid intrusion detection system : Integrating hybrid feature selection approach with heterogeneous ensemble of intelligent classifiers," *International Journal of Network Security*, vol. 20, no. 1, pp. 40–53, 2018.

[23] M. Shen, M. Wei, L. Zhu and M. Wang, "Classification of encrypted traffic with second-order markov chains and application attribute bigrams," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 8, pp. 1830–1843, 2017.

[24] W. Stallings, *Computer and Data Communications Technology*, 2011.

[25] M. K. Steven, *Fundamentals of Statistical Signal Processing Detection Theory*, vol. 2, Prentic Hall PTR, 1998.

[26] Z. Tbatou, A. Asimi, Y. Asimi, Y. Sadqi and A. Guezzaz, "A new mutuel kerberos authentication protocol for distributed systems," *International Journal of Network Security*, vol. 19, no. 6, pp. 889–898, 2017.

[27] N. Ye, "A markov chain model of temporal behavior for anomaly detection," *Information Assurance and Security*, no. 4, pp. 171–174, 2000.

[28] N. Ye, X. Li, Q. Chen, S. M. Emran and M. Xu, "Probabilistic techniques for intrusion detection based on computer audit data," *IEEE Transactions on Systems, Man and Cybernetics Part A:Systems and Humans*, vol. 31, no. 4, pp. 266–274, 2001.

[29] N. Ye, S. Member, Y. Zhang and C. M. Borror, "Robustness of the markov-chain model for cyber-attack detection us air force office of scientific research," *IEEE Transactions on Reliability*, vol. 53, no. 1, pp. 116–123, 2004.

[30] X. You and L. Zhang, "Improved authentication model based on kerberos protocol," *Advances in Intelligent and Soft Computing*, vol. 128, no. 2, pp. 593–599, 2011.

[31] C. Zhou, S. Huang, N. Xiong, S.-H. Yang, H. Li, Y. Qin and X. Li, "Design and analysis of multimodel-based anomaly intrusion detection systems in industrial process automation," *IEEE Transactions on Systems, Man and Cybernetics: Systems*, vol. 45, no. 10, pp. 1345–1360, 2015.

Biography

Fadi Al-Ayed received his B.S. degree in 2010 and his M.S. degree in 2012, both in Computer Science from Arizona State University, USA. Currently, he is pursuing his Ph.D. degree in Computer Science at The Catholic University of America. His current research interests include Compute Network and Security, System Security, Data Retrieval and Data Mining.

Chunqiang Hu received his M.S degree and first PhD degree in computer Science and Technology from Chongqing University, Chongqing, China, in 2009 and 2013, respectively. He obtained his B.S. degree in Computer Science and Technology from Southwest University, Chongqing, China, in 2006. He was a visiting scholar at The George Washington University from Jan., 2011 to Dec., 2011, and then got his second PhD degree in Computer Science, The George Washington University, Washington DC in 2016. He won the Best Paper Award in ACM PAMCO 2016. He is a postdoctoral researcher in The Catholic University of America. His current research interests include privacy-aware computing, big data security and privacy, wireless and mobile security, applied cryptography, and algorithm design and analysis. He is a member of IEEE and ACM.

Hang Liu joined the Catholic University of America as an Associate Professor in the Department of Electrical Engineering and Computer Science in 2013. Prior to joining CUA, he had more than 10 years of research experience in networking industry, and worked in senior research and management positions at several companies. He also led several industry-university collaborative research projects. He was an adjunct professor of WINLAB, ECE Dept., Rutgers University from 2004 to 2012. Dr. Liu was an active participant in the IEEE 802 wireless standards and 3GPP standards. He was the editor of the IEEE 802.11aa standard and received an IEEE recognition award. He was also the rapporteur of a 3GPP work item. Hang Liu received a Ph.D. degree in Electrical Engineering from the University of Pennsylvania. His research interests include wireless communications and networking, Internet of Things, mobile computing, future Internet architecture and protocols, content distribution, video streaming, and network security.