Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 89 (2016) 152 – 161

Procedia Computer Science

Twelfth International Multi-Conference on Information Processing-2016 (IMCIP-2016)

Risk Rating System of X.509 Certificates

Varsharani Hawanna[a,*], V. Y. Kulkarni[a], R. A. Rane[a], P. Mestri[b] and S. Panchal[b]

[a] MAEER's MIT, Pune 411 038, India
[b] IBM Security, IBM, Pune 411 057, India

Abstract

X.509 certificates enable to affirm the distinguishing proof of the parties involved in the communication. As of now, majority of individuals and communities are using X.509 certificates to demonstrate their ID during on-line exchanges, so the unwavering quality and risk level of these certificates come into a question. Hence, we have proposed a framework which assesses risk associated with X.509 Certificates with the help of certain trust criteria and characteristics. For evaluating risk related with certificate we use classification Technique with machine learning algorithm, which categorizes risk of certificate in three levels-High Risk, Medium Risk and Low risk. Our system is useful to find out risk associated with certificate while user carries out important transactions. User needs to input the certificate and system will provide risk associated with that certificate and if it is a high risk or medium risk certificate it will mention due to which parameter it bears risk. Our framework has application in browser-server communication and detecting phishing websites which have Https URLs.

Peer-review under responsibility of organizing committee of the Organizing Committee of IMCIP-2016

Keywords: Certificate Path Validation; Certificate Practice Statement; X.509 Certificate; X.509 Trust Model.

1. Introduction

X.509 certificate[10] or public key certificate is a digitally signed statement that binds the value of a public key to the identity of the individual, gadget, or service that holds the respective private key. One of the principle advantages of certificates is that, it no longer needs to keep up an arrangement of passwords for individual subjects which should be verified as a necessary step to get to, rather, the host only sets up trust in a certificate issuer.

Certificates can be utilized for:

1. Verification, which confirms the identity of an individual or thing.
2. Protection, which guarantees that data is just accessible to the target group.
3. Encryption.
4. Digital signatures.

*Corresponding author. Tel.: +91 -8237248836.
E-mail address: hawanna.varsha@gmail.com

Peer-review under responsibility of organizing committee of the Organizing Committee of IMCIP-2016
doi:10.1016/j.procs.2016.06.027

These are required for the security of your correspondences. Likewise, numerous applications use certificates, for example, email applications and Web programs. Our system has application in Web programs.

In X.509 Public Key Infrastructure (PKI), Certificate Authority (CA) issue a certificate, this certificate is used as recognizable proof of its subject. CA is a trusted outsider between two correspondence parties, which plays the role of issuing certificate by affirming both the parties incorporated into the communication. Generally, the CA issues a certificate to the subject in light of its own rule archives. For example, the Certificate Policy (CP) and Certification Practice Statement (CPS). The CP defines a set of standards that the CA keeps up during the life cycle of a certificate[10]. This is one of the essential documents to quantify the trust level of a certificate.

As of now, there is a well-known approach for managing trust in CAs that comprises a list of trusted CAs in web browsers. This list of root CAs differs from application to application and from OS to OS. Browsers confirm the trust of leaf certificate in light of its CA, but consider the possibility that CA in the trust store no longer be trusted. Recently, Google introduced an article-"Google calls out certificate authorities that can no longer be trusted"[9]. Article discusses about Google's Submariner which fills the gaps of listing certificate authorities that were once trusted, by withdrawing it from Google's root program and including new certificate authorities that are in the pipeline but have not yet been added to the trusted list of root store. So in our framework we considered just single certificate different checks, it is not fully dependent on its CA. Refer section 8 for points of interest.

Currently, there is no automated mechanism for finding the risk level of a certificate. In this paper, we have contributed in proposing such a framework which will automatically evaluate risk associated with certificate, where user needs to input the certificate. We are using Random Forest machine learning algorithm due to its accuracy and efficiency feature. We first collect both trusted and untrusted certificates. Then extracting certificates various fields and storing them into CSV file, which becomes dataset of our system. Now, apply the classifier or algorithm and get result in three categories of Risk. Here machine learning algorithm will be very useful in following scenario: Gradually with time, if value of some attributes for risk criteria changes, just make changes in dataset, our model will automatically learn and give appropriate results.

Overall paper is organized in nine sections. Section 1 gives a brief Introduction of x.509 certificates and problem statement. Section 2 describes about background theory which emphasizes on some primary fundamentals of x.509 certificate. Section 3 explain what is the motivation for us to propose this problem statement. Section 4 gives Extensive Literature Survey of previous work. In Section 5, the Problem definition of the project is discussed. Section6 gives description of system. Section 7 shows mathematical model for proposed approach. Section 8 gives the implementation details of modules. Section9 discusses the conclusion and future works.

2. Background Theory

X.509 was first issued in 1988 and was started in association with the X.500 standard. It strictly follows structural system of certificate authorities (CA) for issuing the certificates. The X.500 framework has ever been implemented by sovereign countries for state identity information sharing settlement satisfaction purposes, and the IETF's Public-Key Infrastructure (X.509), or PKIX, Experts have adapted the standard to the more efficient association of the Internet. Actually, the term X.509 certificate more often refers to the IETF's PKIX Certificate and CRL Profile of the X.509 v3 certificate standard, as determined in RFC 5280[10], usually alluded to as PKIX for Public Key Infrastructure (X.509).

In the following subsections we will see standard format of X.509 all versions certificate, its sample certificate, and file format or extensions of X.509 certificate. Further from this point, in this paper we will use the term certificate which refers to X.509 certificate

2.1 Standard format of X.509 certificate

Following figure demonstrates the standard format of all forms of x.509 certificate. Version1 incorporates essential fields, Version 2 incorporates all fields of version 1 and in addition to it issuer unique identifier and subject unique identifier fields, yet it is not generally utilized as a part of the protected exchanges on Internet. Therefore, version 3 accompanies additional extension fields.

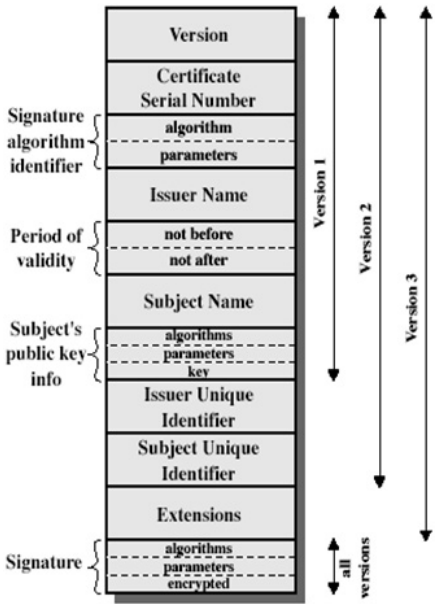

Fig. 1. Standard Format of x.509 Certificate.

Certificate field description:

Version: 1, 2, 3 are the three versions of X.509 certificate structure. V3 certificates have wide utilization.

Serial Number: It is a unique number assigned by CA to the subject.

Signature Algorithm: Used by the CA to sign the certificate. It can be RSA + MD5, RSA, MD5.

Issuer Name: This field indicates information with respect to the CA that issues certificate.

Validity Period: It determines begin and end date of the certificate. Certificate must be in its legitimacy period.

Subject Name: specifies the name of individual, PC, gadget, or CA to whom certificate is issued.

Subject Public Key Info: specifies about public key type and length associated with the certificate.

Subject/Issuer Unique Identifier (v2only): It is an optional field. It provides a location to specify a bit string to uniquely identify the subject/issuer name, in the event that the same name has been assigned to more than one subject/CA over the time.

Extensions: This field is present in x.509 v3. Fields in extensions are not same in all the certificates of version 3. If they are available, will be shown. For additional extension fields refer[5].

2.2 Sample X.509 certificate

We are demonstrating ICICI bank certificate which is issued by VeriSign Certification Authority. Double tap this certificate to view its properties and details. This data is shown on three tabs: General, Details, and Certification Path. First we will see general tab, it contains certificate information about purpose for which the certificate is intended, issuer to, issued by, validity period as shown in Fig. 2(a). One can Install certificate, convert it in base 64 format and save in their computer.

Detail tab contains information as shown in figure Fig. 2(b). In Detailed Tab when one clicks on copy to file, it will lead to Certificate Export Wizard. It will help to copy file in your computer, in Der, Pem (base64), .PKCS#7, .PKCS#12 file formats. Der and pem format have .cer as extension, PKCS#7 and .PKCS#12 have .p7b and .pfx extension respectively. Certification Path tab, shows hierarchy of leaf certificate.

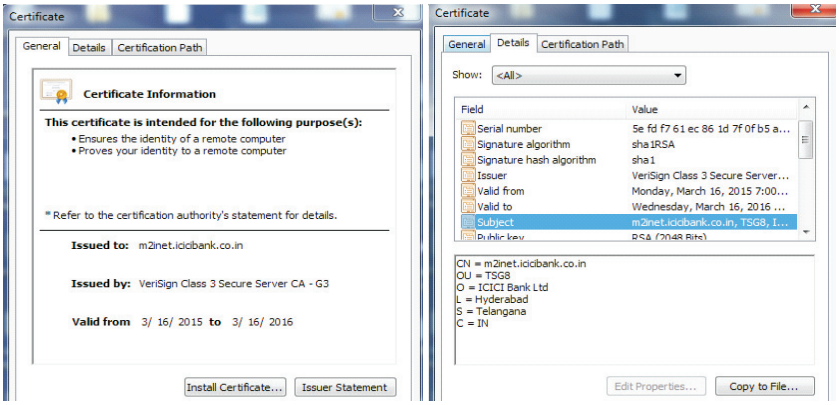

Fig. 2. (a) General Tab of Certificate; (b) Detail Tab of Certificate.

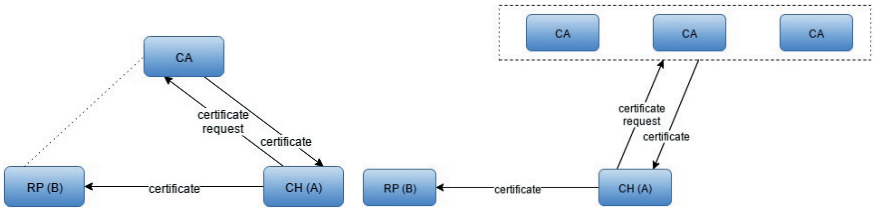

Fig. 3. (a) Previous Scenario of x.509 Certificate Trust Model; (b) Current Scenario of x.509 Certificate Trust Model.

3. Motivation

The X.509 trust model has three primary substances: certification authority (CA), certificate holder (CH) and relying party (RP). The certification authority is a trusted outsider between the certificate holder and relying party. CAs primary role is to issue a certificate by confirming both the inputs included in the communication. If the certificate holder needs to correspond with RP, CH should give the affirmation of identity, so certificate holder request for a certificate to CA. CA issues a certificate to certificate holder by verifying certificate holder information and by analyzing various guideline documents such as the Certificate Policy (CP) and Certification Practice Statement (CPS).

Anyway, this methodology would be effective if there is only CA present or if RPs had a previous association with the CA,as shown in Fig. 3(a). But currently, circumstance is altogether different from previous one. Nowadays, there are many CA established so RPs have no connection with any CAs at all, as appeared in Fig. 3(b).

Subsequently, RPs need to fabricate their trust choice by performing a few checks: the signature on the certificate must be confirmed, path from the certificate to a trusted root certificate must be found and assessed, and the extension fields must be checked, so on. Another most essential thing that RP needs to analyze is a set of documents like Certificate Policy (CP), Certification Practice Statement (CPS). These are the critical steps to verify the trust level of certificate or discovering risk level connected with it.

To help RP, we propose a framework which automates all tasks, discovers the risk level of certificates and passes onto RP. In our system user is the RP, server is the CH when certificate is new or unknown to the browser. System plays intermediate role between RP and CH.

4. Literature Survey

There are many authors who propose their work to help RP in evaluating the trust of certificate.

Wazan A.S. propose an explicit master recommender[15] whose action is to give the fundamental data permitting RPs to settle on an informed choice around a CA. Now, the RP comprehends whether to accept a certificate or not for

Table 1. Literature Survey Table.

SN	Paper Title	Publication	Features	Gap Identification	References
1	The X.509 trust model needs a technical and legal expert.(2012)	IEEE	Solution used in both situations when RP knows CA and when RP don't know about CA.	Find trustworthiness of certificate based on its CA.	[15]
2	A formal model of trust for calculating the quality of X.509 certificate.(2011)	IEEE	While accepting certificate users can specify risk they want to take.	To find out trust of certificate System requires more Calculation.	[13,16].
3	A Model for Automatically Evaluating Trust in X.509 Certificates.(2010)	Cybernetica	Provide a ternary recommendation regarding certificates trustworthiness, not binary.	Users can manually add any known or obscure certificates to the data store and utilizes this for discovering trust.	[1].

a specific transaction. The master recommender ought to be autonomous of PKIs and must assume both the part of technical and legal expert for helping the RPs. By including explicitly this role to X.509 trust model, the assignment of RPs is streamlined, and RPs has to depend just on the master and not on every single CA of certificate holders.

In paper[13,16], wazan, labored, barrere and benzekri proposed a structure which gives RPs qualitative data to decide the Quality of Certificate (noted QoCER). This quality is ascertained in light of two parameters: QoCPS and QoPKI. Relation between QoCER, QoPKI and QoCPS as following: QoCER $= \varphi$(QoPKI) X QoCPS; where function φ can be any function that outputs a value between 0 and 1, QoPKIlies between 0 and 1, QoCPS lies between 0 and 1.

Authors propose a procedure of acceptance which incorporates the idea of QoCER. First, structure performs basic search for the certificate of root authority. In the event that this certificate has a place with the static trust domain, the RP accepts the certificate. The static trust area contains completely known and controlled CAs. This allows improving the performance of the acceptance procedure when managing with well-known partners. Otherwise, the RP processes the QoCER that comprises of: a) Getting the QoCPS from an authority which is recognized to perform this assignment; b) Getting the QoPKI of the CAs that assigns this certificate from an authority recognized to perform this task; c) Calculating the QoCER as per the specific function φ; This quality speaks to a quantitative evaluation of the authenticity and trustworthiness of the certificate.

In paper[1] Ahmed and Bogdanov's Framework executes as follows: Initially, the customer application keeps up a data store of trusted and untrusted certificates. when a customer application needs to assess a certificate, it 1st checks in its own data store. If the certificate is in the trusted zone, it is acknowledged, otherwise, rejected. Clients can manually add obscure certificates to the data store. However, if the certificates status can't be assessed from the data store, framework performs extra confirmation steps to consequently assess the trust level of a certificate. 2nd the framework can check if the key utilization field matches with the usage requirements of the customer application. If yes, then proceed with the assessment. Otherwise, the certificates ought to be rejected. 3rd, it checks amount of data accessible in the certificate's distinguished name (DN) field. Certificates with less attribute data get a lower rating. In the 4th step, the application checks the accessibility of the CPS link field. A certificate without a CPS link is considered as a low trustworthy. In the 5th step, the application can utilize the semi-formalization technique on the CPS to assess the trust level of a CA. A rating is given in light of all the above 5 steps assessment and threshold values.

We identified some features and gap identifications of above papers and showing this in tabular format of Table 1.

There are some authors who used classification on certificate datasets for different purposes. So we did brief literature survey of such papers and described as follows:

Mishari, present a novel strategy to recognize web-extortion by recognizing domains hosting such as malignant exercises[11]. Authors collect certificates of authentic domain from Alexa site, popular domains from the .com/.net Internet Zone File and those utilized by deceptive from different domains of phishing URLs. Authors examined these collected certificates and think of components which uncover contrasts between certificates used by deceptive and authentic/popular domains. Authors use classification technique for Identifying Web-Fraud. We studied these features and utilize some of them for our risk analysis.

Zheng Dong[6] proposed a machine-learning way to deal with distinguished phishing sites utilizing features from X.509 public key certificates. It utilizes Planet Lab for certificate gathering from three landmasses. Authors framework,

first download certificate from given site, once it is downloaded, it is parsed locally into a set of values as indicated by the standard X.509 certificate fields. The 42 elements are then computed from the mandatory and optional fields. The instances are then spared in an ARFF format and send it for further classification. We get some thought by concentrating on various features which is gathered by parsing X.509 certificate fields.

5. Problem Definition

By doing brief literature survey and studying different features and finding out gap identifications of some initial research, we build our problem definition. We isolate it in two phases as follows:

Phase I: Rate the risk level considering different traits and attributes of certificate using classification techniques.

Phase II: Our framework has an application in browser and server communication during SSL handshake. We will implement this framework system as a plug in for browser, hence when certificate is unknown/new to browser our system will track it, find out risk and finally display risk associated with certificate on screen so user can decide whether to continue a transaction or not, by considering risk factor.

6. Proposed System

As depicted in section 5, our framework has two stages. In stage I, we execute our framework as stand alone software, so user can enter certificate and find out risk associated with it. After completing Stage I, we will write a plug-in for browser. We divide our framework in three modules.

- Module 1 involves collection of trusted and untrusted x.509 certificate and store in trust store.
- Module 2 involves collection of trust attribute/criteria.
- Module 3 deals with implementation of classification.

In this paper we portray our work with respect to stage I. As depicted in Fig. 4, any user inputs the desirable certificate to find out the risk associated with it. Our classification module applies certain trust criteria; according to matching attributes it calculates result in % matching and classifies risk in three levels – high, low, medium risk. We set a Medium risk (MR)_threshold and Low risk (LR)_threshold according to trust attribute/criteria weight or impact. The system compares results with LR & MR_threshold and classify accordingly as shown in Fig. 4.

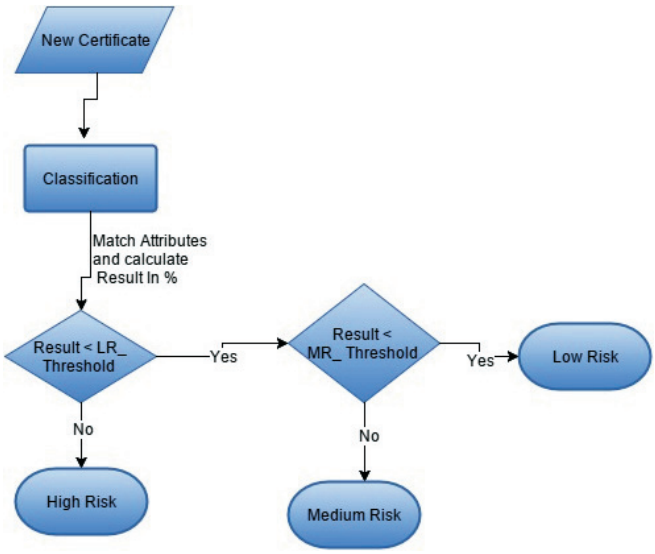

Fig. 4. Flow of Proposed Framework.

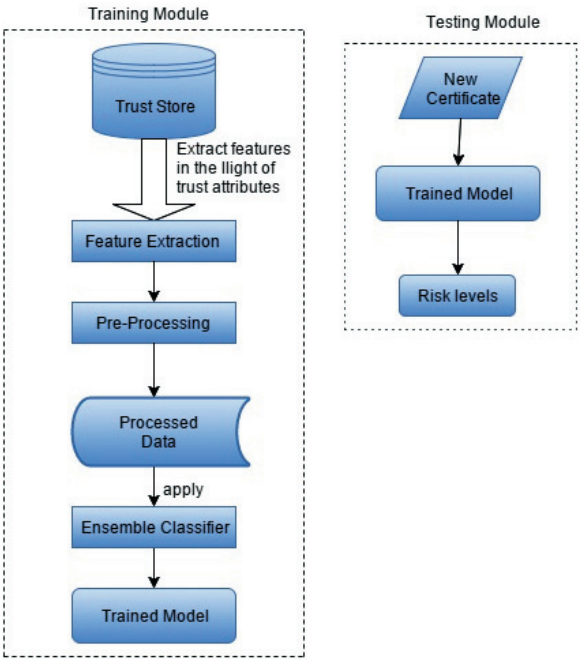

Fig. 5. System Architecture with Classification Module.

Our system first gathers trusted and untrusted certificates and store in trust store, as shown in Fig. 5. (See section 8). Each certificate contains attributes as appeared in section 2.1 sample certificates detail tab. Then we extract only those attributes which are considered in the trust assessment process, (see list of attributes in section 5). We extract these features (attributes), using java libraries like java.security.cert.X509Certificate.

As described above version 3 certificates contain extension fields; however, these fields are not same for each version 3 certificates so we require preprocessing on the extracted feature. In preprocessing we fill missing values of extension fields and get processed data in the format which will require for the classification.

We have chosen to utilize R programming as a classifier device due to its usability, built in packages and Random forest (RF) as an ensemble classifier due to its effectiveness and more accuracy features. Another reason to use RF is that it has a feature to find out most important attributes.

7. Mathematical Model

Following is the mathematical model of certificate Risk Rating System (CRRS):

CRRS $= \{I, F, O\}$; where, $I =$ Input set; $F =$ system function; $O =$ output set.
$I = \{C_{u1}, C_{u2}, C_{u3} \ldots C_{un}\}$; where, $C_{u1}, C_{u2}, C_{u3} \ldots C_{un}$ are the n no of certificates.
$O = \{H_R, M_R, L_R\}$; where, $H_R =$ High Risky; $M_R =$ Medium Risky; $L_R =$ Low Risky.
$F = \{F_{DC}, F_{FE}, F_{DSS}, F_C\}$; where, $F_{DC} =$ Data Collection; $F_{FE} =$ Field Extraction; $F_{DSC} =$ Data SetCreation; $F_C =$ Classification.

Now, we will see each function in detail.

1) $F_{DC} = \{W_s, N_{mt}\}$; where, $W_s =$ Wireshark tool; $N_{mt} =$ Network Miner tool. Now, $W_s = \{W_{PL}\}$; where, $W_{PL} =$ Winpcaplibrary. $W_{PL} =$ {*pcap_lookupdev(), pcap_setfilter(), pcap_lookupnet(), pcap_sendpacket()*}; where, *pcap_lookupdev()* = to extract the list of all the adapters; *pcap_setfilter()* = apply a filter; *pcap_lookupnet()* which retrieves IP address and mask of a specified network adapter; *pcap_sendpacket()* = The function used for sending packets.

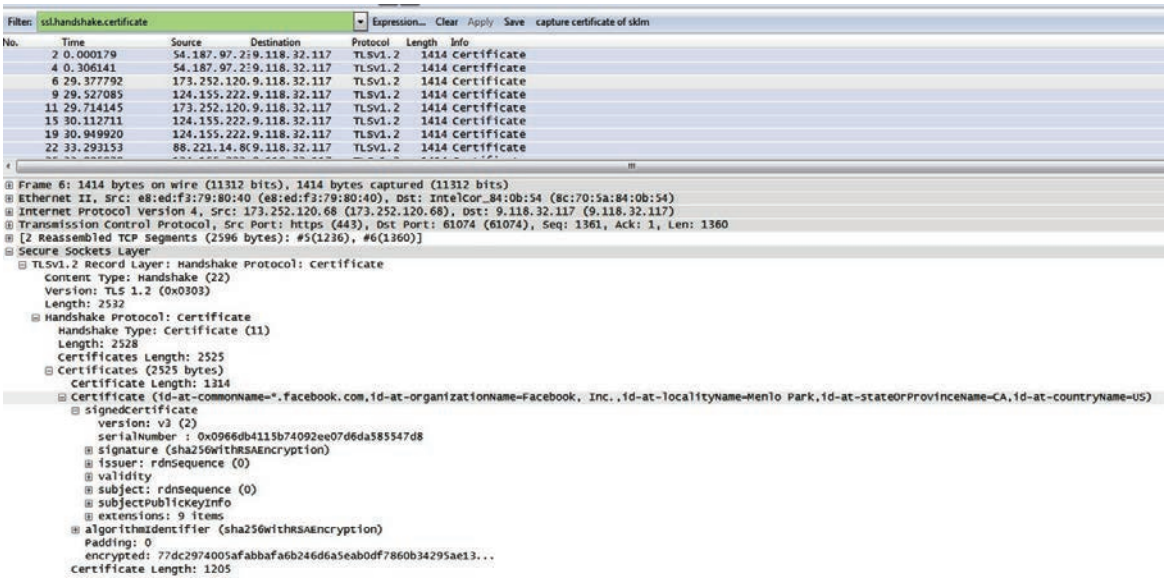

Fig. 6. Wireshark Captured Sample PCAP file.

2) F_{FE} = Field Extraction $\{J_F\}$; where J_F = Java Function; $J_F = \{H\}$ where, H = Hashing {**hashCode**()} Hash code $h(s)$ defined by $h(s) = \sum_{i=0}^{n-1} s[i] \cdot 31^{n-1-i}$ where, terms are summed using Java 32-bit integer addition, $s[i]$ denotes the ith character of the string, and n is the length of S.
3) $F_{DSC} = \{F_{FE}, DS_{CSV}\}$; where, F_{FE} = Feature Extracted from above module; DS_{CSV} = Data set in CSV format; by using Java F_{FE} are inserted into CSV file.
4) $FC = \{DS_{CSV}, RF_C, R_P\}$; where, RF_C = Random Forest Classifier; R_P = R Programming. Apply RF_C on DS_{CSV} using R_P as a classifier tool.

8. Implementation Details

Module 1: Data collection:

For the collection of trusted certificates, we use Alexa[2] https domains. Alexa is a standard website relevant for its Internet websites positioning according to most well-known or most utilized sites as a part of the world. It also provides a facility of narrowing the search by country, by category like news, games, etc. From these we tried top 900 ranked websites sorted by nation India. As a result, we were able to gather near about 1902–2100 certificates using wire shark[12] and network miner[8] tool, out of which we found nearly 2000 certificates unique.

Initially, we go on alexa site, copy a legitimate URL with a constraint that it should be https and paste in browser. In background we run wire shark, when https sites exchange certificates during SSL handshake, it captures traffic of these domains on port 443 which is standard port of https protocol, with " ssl.handshake.certificate" as a filter.

It generates a PCAP file, which comprises of number of packets and each packet with this filter contains five frames-Frame, Ethernet, Internet Protocol, Transmission Control Protocol, secure socket layer. From these frames certificate is stored in secure socket layer frame, as shown in Fig. 6.

Then we feed this PCAP file as an input to the network miner as shown in Fig. 7, which extracts certificate from packet and stores in different folders. We collect these certificates from all folder and store in data store; we call this as trust store.

Now, for the collection of untrusted certificates we utilize phishing URL from phishtank.com site, it contains rundown of phishing URLs. By applying past strategy, currently we get almost 1200 certificates out of which 900 are unique. For both trusted and untrusted certificates our capturing is still in further progress.

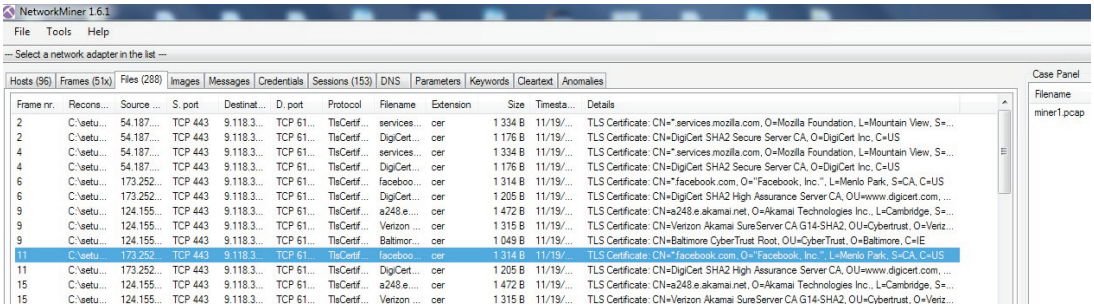

Fig. 7. Network Miners Extracted Certificates.

Module 2: Attributes and criteria used for finding the risk.

Assumption: risk and trust are opposite fields, if trust is 1 out of 10 then risk will be 9 out of 10.

- End date > current date: when user inputs the certificate, we compare its end date and current date. End date of certificate should be greater than current date. We perform comparison by converting date into no of days. If certificate gets expired i.e. end date is less than the current date, we label this certificate as high risk.
- Serial No and CRL: Certificate Revocation List (CRL)[10] is a list of serial nos., that have been revoked by certification authority for following reasons-Superseded, Cessation of Operation, Affiliation changed. We first download this CRL, which will be available as CRL Distribution Point field. By extracting CRL Distribution Point field which gives value as URL from where we can download the CRL list. Then extract serial no. from certificate and check whether this serial no. is listed in CRL, if it is found in CRL list it will be directly treated as high risk certificate. If not found in CRL list it will support trust criteria.
- Domain name and subject name should be the same, this leads to positive effect in trustworthiness (low risk)[3].
- CPS link: Certificate Practice Statement (CPS)[7] is an archive from a certificate authority which portrays their practice for issuing and managing certificates. Presence of CPS link in its extension field, leads to positive contribution in less risk (trustworthiness).
- CPS document: Clicking on cps link redirects to a webpage, in this webpage first link contain a CPS document. This documents size is different for different CA. Presence of cps document on link will be treated as positive contribution in trust (low risk).
- CPS weight: we parse the cps document and verify all the 27 attributes described by authors Omar Batarfi, Lindsay Marshall in paper[4]. We rate according to matching attributes and treat this as one parameter in our calculation. We decide to build our own algorithm to parse the CPS document using natural language processing.
- Chain validation: For each CA certificate we validate the trust. We perform following sub-checks for chain validation:

  1. CA should have more validity than its leaf certificate.
  2. Issuer name of leaf certificate should be same as subject name of CA.
  3. If it is a CA certificate, CA bit must be 1.
  4. If CA certificate has path length = 0 it can issue only leaf certificate. Refer[3].

- Large key size, leads to contribution in more trust. Here, we are considering key size with respect to its public key algorithm.
- Signature algorithm should be strong. Only MD5 leads to positive contribution towards high risk[11].
- Certificate should be in its valid period. Not more than decade, not less than a year[11]
- Subject and issuer name shouldn't be some state, some organization. It should have valid names. If these fields contain invalid names or valid names but less information, it leads to positive contribution towards high risk[3].

Hence we are validating 12 fields and 31 sub-fields i.e. total 43 attributes or criteria of a single certificate for rating the risk.

9. Conclusions and Future Work

Our system is undertaken to find out risk level of x.509 certificate as high risk, medium risk and low risk form. For this we are using classification with performing certain checks and trust criteria. For classification we decided to use Random Forest algorithm and *R* programming tool. As an application, we will write our system as a plug-in for browser so, it will help user for knowing risk associated with certificate for further transaction, when certificate is not known or new for browser. In this paper, we described about how we collect the certificates and list many trust criteria/attributes which we are considering during risk level calculation.

In future we will extend our system by adding different features of legitimate website for detecting real-time phishing attacks both Http and Https based.

Acknowledgement

We want to express gratitude toward Mr. Mahesh Paradkar, Senior project Manager of IBM India, Rick Robinson, Offering Manager of Data security IBM India and my team for prompting me on this point.

References

[1] A. S. Ahmed and D. Dan Bogdanov, A Model for Automatically Evaluating Trust in X.509 Certificates, *Cybernetica Research Report*, pp. 12–16, (2010).
[2] Alexa.com. The Web Information Company. http://www.alexa.com
[3] C. Brubaker, S. Jana, B. Ray, S. Khurshid and V. Shmatikov, Using Frankencerts for Automated Adversarial Testing of Certificate Validation in SSL/TLS Implementations, In *2014 IEEE Symposium Security and Privacy (SP), IEEE*, pp. 114–129, May (2014).
[4] O. Batarfi and L. Marshall, Defining Criteria for Rating an Entity's Trustworthiness Based on its Certificate Policy, In *Proceedings of the First International Conference on Availability, Reliability and Security (ARES'06), IEEE*, p. 8, April (2006).
[5] I. Curry, Version 3 X, 509 Certificates, *Entrust Technologies*, (1996).
[6] Z. Dan Dong, A. Kapadia, J. Blythe and L. J. Camp, Beyond the Lock Icon: Real-Time Detection of Phishing Websites Using Public Key Certificates, In *2015 APWG Symposium on Electronic Crime Research, IEEE*, pp. 1–12, May (2015).
[7] W. Ford and S. Chokhani, CygnaCom, Inc., VeriSign, Inc. Certificate Policy and Certification Practices Framework, *RFC 2527*, March (1999).
[8] A. Ghafarian, An Empirical Study of Network Forensics Analysis Tools, In *ICCWS2014-9th International Conference on Cyber Warfare & Security*, pp. 366, March (2014).
[9] Googles Article, Google Calls Out Certificate Authorities that can No Longer be Trusted, *Infoworld*, 28 March (2016).
[10] R. Housley, W. Polk, W. Ford and D. Solo, Internet X. 509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, *RFC 5280*, (2002).
[11] M. A. Mishari, D. E. Cristofaro, K. E. Defrawy and G. Tsudik, Harvesting SSL Certificate Data to Identify Web-Fraud, arXiv preprint arXiv:0909.3688v4 [cs.CR], pp. 1–13, 13 January (2012).
[12] C. Sanders, Practical Packet Analysis: Using Wireshark to Solve Real-World Network Problems, *No Starch Press*, 2nd edition, (2011).
[13] W. A. Samer, L. Romain and B. Francois, A Formal Model of Trust for Calculating the Quality of X, 509 certificate, *Security and Communication Networks*, pp. 651–665, (2011).
[14] G. A. Weaver, S. Rea and S. W. Smith, A Computational Framework for Certificate Policy Operations, In *Public Key Infrastructures, Services and Applications; Springer Berlin Heidelberg*, pp. 17–33, (2009).
[15] A. S. Wazan, R. Laborde, F. Barrère and A. Benzekri, The x. 509 Trust Model Needs a Technical and Legal Expert, In *2012 IEEE International Conference on Communications (ICC)*, pp. 6895–6900, June (2012).
[16] A. S. Wazan, R. Laborde, F. Barrère, A. Benzekri, X. Validating, 509 Certificates Based on their Quality, In *ICYCS 2008, The 9th International Conference for IEEE Young Computer Scientists, 2008*, pp. 2055–2060. pp. 281–304, November (2008).