SN Computer Science (2022) 3:152
https://doi.org/10.1007/s42979-022-01048-6

SURVEY ARTICLE

Network Communication Encoding: A Study for Authentication Protocols

Tejaswini Apte[1] · **Priti Kulkarni**[1]

Received: 17 August 2021 / Accepted: 25 January 2022 / Published online: 3 February 2022

Abstract

The use of internet has increased significantly in the COVID-19 pandemic, and this has set the ground for various cyber-attacks, which are executed over the network during data transmission. This scenario is proven to be multifold for accessing the cloud remotely deployed in university premises. To provide secure authentication and compatibility over heterogeneous systems for cloud accessibility, every network communication applies an encoding scheme to standardize data transmission. With many wireless and ad-hoc networks where the nature of communication is difficult to monitor, the encoding scheme prevents malicious code injection during data transmission. The objective of this paper is to study encoding schemes available for data transmission and their application in terms of authentication protocols such as Kerberos and LDAP. Furthermore, it will also emphasize on the design of integration model of Kerberos and LDAP to Cloud and Shared Storage to evaluate the impact of ASN.1 vulnerability.

Keywords Authentication protocols · ASN.1 · Encoding · TLS · Vulnerability · Cloud · Shared storage

Introduction

The authentication protocols enable secure data transmission over the network. The objects in every network have an entry in Management Information Base (MIB), which is a set of network objects managed by Simple Network Management Protocol (SNMP) [1]. There were compatibility issues revealed in the literature during data transmission by SNMP with respect to incompatible data type encoding. To minimize the incompatibility during encoding, Abstract Syntax Notation One (ASN.1) defines a data structures used for serialization and deserialization in a cross platform deployment for example, cloud deployment [2]. Every network presentation layer is subject to use Abstract Syntax Notation One (ASN.1) to exchange data among various devices over the network. The presentation layer initially acts to define generic structure of data, which is later followed by concrete syntax according to local language of system. Next, the transfer mechanism defines the data representation along with encoding method. The layer then forwards encoding and decoding rules to application layer to translate the encoded data accordingly [3].

The role of ASN.1 allows the sending and receiving of data in a device-independent format, i.e., independent of architectural conventions of the sender and receiver or in cross platform deployment [4]. To support the cross platform encoding deployment or a respective application, ASN.1 uses three different methods for encoding based on type and length of values. The methods are (a) primitive definite-length method, (b) constructed definite-length method, and (c) constructed indefinite length method. The primitive method requires length to be known in advanced and implemented for simple types. The next method, constructed definite-length method, can be applied to simple and structured data types and also derived data types. It requires length of data or value to be known in advanced. Finally, the constructed indefinite length method applies to simple

"This article is part of the topical collection "Cyber Security and Privacy in Communication Networks" guest edited by Rajiv Misra, RK Shyamsunder, Alexiei Dingli, Natalie Denk, Omer Rana, Alexander Pfeiffer, Ashok Patel, and Nishtha Kesswani".

✉ Priti Kulkarni
pritiap@gmail.com

Tejaswini Apte
apte.tejaswini@gmail.com

[1] Symbiosis Institute of Computer Studies and Research (SICSR), Symbiosis International (Deemed University), 1st Floor, Atur Centre, Gokhale Cross Road, Model Colony, Pune, Maharashtra 411016, India

and complex data types. The length of the data types is not required to be known in advance [5]. In addition to these methods, ASN.1 follows certain rules for certificate generations, digital signatures, and preservation of encoding. The said rules are named as Basic Encoding Rules (BER), Distinguish Encoding Rules (DER), and Canonical Encoding Rules (CER), respectively. BER is a composition of Type, Length, and Value (TLV structure) which is responsible to generate certificate for successful handshake [6]. DER is accountable to provide encoding for the digital signature and cryptography. CER encoding focuses on preservation of encoding, this is required in security exchanges [7].

Many of the protocols such as Transport Layer Security (TLS), Kerberos, and Lightweight Directory Access Protocol (LDAP), etc. rely on ASN.1 BER, which is subsequently discussed in the section "Literature Review". Every authentication protocol has an application, where the vulnerability is reported and a solution is suggested. The section "Methodology" narrates the process of evolving the ASN.1 in Kerberos and LDAP, along with the use case, i.e., cloud and respective shared storage. We have also presented Infrastructure model of cloud and shared storage in the section "Methodology". Finally, the conclusion and future direction is presented in the last section.

Literature Review

Transport Layer Security (TLS) protocol provides certificate-based authentication to manage secure communication among two nodes. This protocol has handshake and record layer for establishing the sessions and exchanging the messages. The specific nodes maintain the status and state of reading and writing while communicating over the network. The handshake layer contains the security parameters, i.e., certificate information before the message can be transmitted. The message then divided into number of fragments. Each fragment is transmitted with the security parameters specified by handshake layer. It uses asymmetric key cryptography to generate public and private key pair to verify the identity during communication. The RFC5878 [8] documentation shows authorization extension to TLS handshake protocol. The extension is introduced in TLS handshake layer to enable TLS for exchanging authorization information between nodes before generation of any authentication certificate. The certificate generated during handshake is encoded with ASN.1. The error in implementation of such encoding builds a vulnerable platform to execute various attacks such as Denial of Service or Buffer overflow [9]. TLS performs baseline security layer to implement authentication protocols, which is outlined in subsequent paragraphs.

Kerberos is designed to authenticate the subject for a given object over the network. A requestor proves its identity by obtaining a ticket from Kerberos Server. A ticket is being granted to requestor in form of principal, which is a unique identity to assign to each ticket granted. To verify the identity, the principal identifiers are encoded in two parts realm and remainder. The realm specifies the domain along with the components required for principal as remainder. The said domain is responsible to authenticate the user for Kerberos and thereby for respective service. Kerberos principal obtains tickets for authentication and respective authorization. The tickets then contains the attributes of nodes, ticket lifetime, and the session key for the requestor for accessibility over network [10, 11]. There was a lacuna during accessibility or transmission of messages over the network for interoperability. The independent encoding for multibyte quantities was considerable driver for the said lacuna, which was implemented in version 4 of Kerberos. As a result, the transmission among various nodes was constrained, where encoding order may not be understood by receiver. To enhance the transmission thereby interoperability, the standardization for encoding is implemented with ASN.1 BER in Kerberos version 5. Thus, the realm and each component of the remainder are encoded as separate ASN.1 General Strings [12]. ASN.1 encoding minimizes the validation, which was required to understand the semantics for the messages transmitted in diverge context for non-standard encoding [13]. For the heterogeneous deployment, Kerberos authentication monitors various devices compatibility for encoding and ASN.1 provides the standardization to shaft incompatibility, if any. Thus, ASN.1 enables orientation for processing Kerberos over TLS during "Hello Message" and the respective response [14]. Figure 1 illustrates the encoding implementation during communication among nodes.

As depicted in Fig. 1, Node A (Requestor) is requesting an authentication ticket from Node B (Kerberos Server) to access Server Node. Each request and response for ticket generation and subsequent accessibility to the Server Node imperatively use ASN.1 encoding to minimize the validity check during message transmission. As mentioned in aforesaid paragraph, ASN.1 provides standardize encoding, and hence supports for heterogeneous infrastructure deployment of Kerberos. Any irregularity in implementation of ASN.1 has large positive impact on introducing the vulnerabilities. The impact of these vulnerabilities are multiplied when Kerberos is deployed for Cloud. For the vulnerability CVE-2020-28196, i.e., unbounded recursion via ASN.1, a security fix is offered by SUSE SLES12 Security Update for OpenStack cloud deployment [15].

Next, Lightweight Directory Access Protocol (LDAP) an authentication protocol, with major entities to validate the identity, domain and respective directory structure are Distinguish Name (DN), Domain Component(DC), and Organization Unit (OU), etc., respectively [16]. Initially, it was designed for direct mapping to string-based

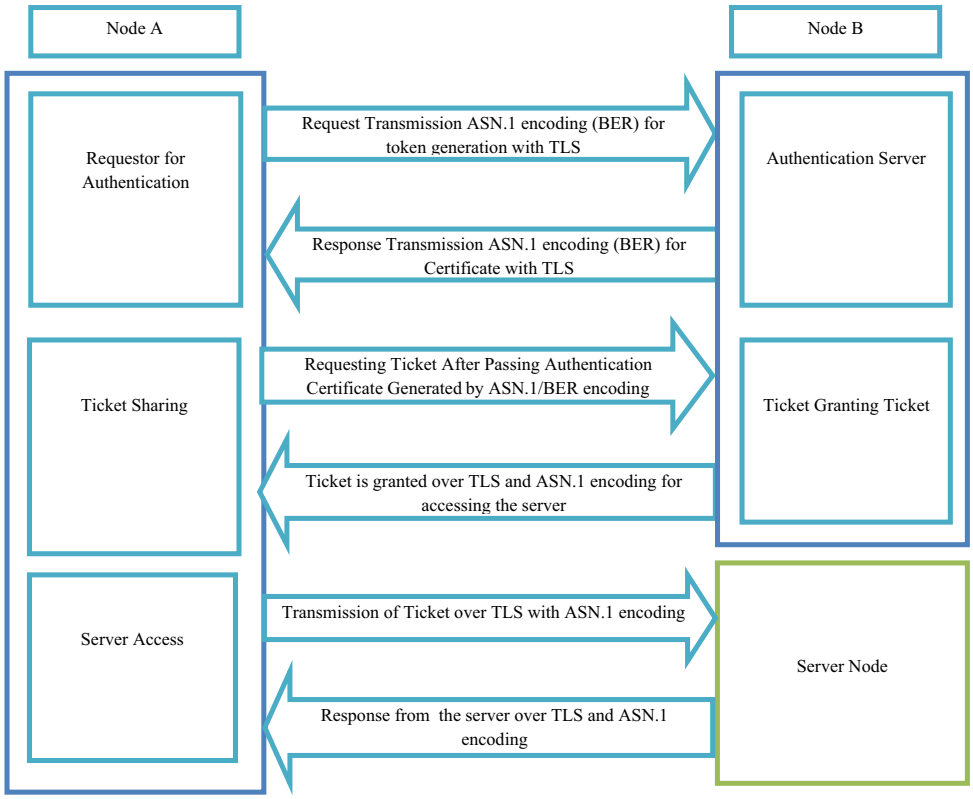

Fig. 1 Communication for authentication with Kerberos via ASN.1 over TLS

encoding of names and attribute values, while transmitting the messages over network. As a result, the scope of LDAP for ad-hoc development of syntaxes and required parsing was limited, which in turn has large impact on certificate generation by Public Key Infrastructure (PKI) for authentication. PKI, a mean to determine the certificate distribution and revocation list, follows ASN.1 standard. This limitation of LDAP was significantly impacting the certificate distribution and revocation process because of poor understanding of ASN.1 encoding standards. As a result, LDAP search was not recognizing ASN.1 types in the definition of the certificate. Generic String Encoding Rules (GSER) was introduced in LDAP to implement new string encoding to retain the structure of the ASN.1 type with existing encoding mechanism, provided that an LDAP server is ASN.1 aware. In 2004, IBM has initiated a project to enable LDAP aware of ASN.1 [17, 18]. The current release of LDAP has extended itself to be aware of ASN.1 and, hence, support PKI certificate implementation process for authentication. LDAP message layer is responsible to manage synchronization of request and response. ASN.1 encoding is embedded into request and response control of nodes in the process of authentication [19]. The start TLS mechanism is an extension of request and response method provided by ASN.1 in LDAP [20]. In Fig. 2, Node A is requesting the validation of authentication to LDAP protocol, and the said request is then forwarded to authentication server to generate certificate. For each communication, the ASN.1 BER is implemented over TLS for maintaining standardization in encoding during transmission.

Though the base support of TLS along with ASN.1 for data transmission was implemented, but there were vulnerabilities reported due to failure in maintaining the ASN.1 rules during implementation. The said vulnerability impacts file sharing capability of Samba Server. As authentication is proven a dominant factor to impact cyber-attacks, the successive section outlines the evolution of ASN.1 in the

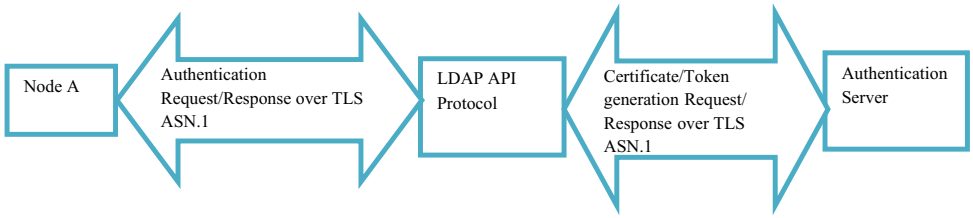

Fig. 2 Communication for authentication with LDAP via ASN.1 over TLS

mentioned authentication protocols along with its application in cloud and shared storage.

Methodology

Every encoding communication standard has a large impact on authentication thereby privacy during network communication among nodes. Various research papers available in literature were scrutinized to visualize the outcome of ASN.1 in Kerberos and LDAP to examine the transmission of data over the network. Table 1 summarizes our findings.

Table 1 presented is conferred the reason about introducing ASN.1 encoding platform for message transmission over the network. The vulnerability discussed in this direction is subject to impact Cisco and Apple-related products. The impact elements to Denial of Service attack due to ASN.1 decoder infinite condition [24]. Samba, a file sharing protocol over the network is designed to use LDAP protocol for authentication. For the LDAP packet size of 13,000 bytes, ASN.1 vulnerability is enough to crash Connectionless Light Weight Directory Access Protocol [23].

The vulnerability mentioned assorted to substantial impact while accessing cloud with the selected authentication protocols. Various approaches are suggested in a literature to integrate OpenStack Cloud with LDAP and Kerberos [25, 26]. However, to this paper, writing the impact of ASN.1 vulnerability to the forenamed integration is required to be explored.

To integrate the relevant protocols with cloud and conceptualize the impact of ASN.1 vulnerability, we are proposing the infrastructure model presented in Fig. 3. OpenStack cloud is deployed with keystone authentication service. This model catalyzes the integration of authentication protocols with keystone to spawn instances and access to shared storage. Authentication to spawn the instances is delivered via Kerberos and Shared storage is accessed by LDAP authentication service. Cloud Identity Service, i.e., keystone, must be extended to encapsulate Kerberos ticket and LDAP certificate to access server. The deployment environment is designed with Intel i7-8700 3.20 GHz (×12 cores) processor with 32 GB RAM. OpenStack Yoga release is installed as a Single Node Installation using DevStack installation. The implementation details of respective model are in progress to understand the said impact.

Conclusion and Future Work

We have examined the importance of encoding scheme for the authentication. The encoding scheme requires a careful design and implementation to minimize the attacks. ASN.1 BER encoding scheme is exercised as a standard practice

Table 1 Necessity of ASN.1 in Kerberos and LDAP

Criteria	Kerberos	LDAP
Encoding schemes	Version 4 was designed with independent encoding; as a result interoperability among multiple nodes was prohibited due to poor understanding by receiver [12]	Initially, LDAP was designed to support string-based encoding to transmit message over the network [18]
Resultant impact	Validation and analysis of semantics of each message is required to minimize the ambiguity [13]	Scope of LDAP for ad-hoc development of syntaxes was limited to address certificate generation specifically PKI. As, PKI is implemented with ASN.1 standard [17, 18]
ASN.1 SUPPORT	To minimize the validation and introduced the standardization among encoding ASN.1 was introduced in Version 5 of Kerberos [13]	Encoding rules such as GSER, was introduced to have LDAP string encoding to preserve ASN.1 semantics, provided LDAP should be ASN.1 aware. IBM has initiated the project to implement awareness of ASN.1 in LDAP [18]
TLS communication	To have a secure communication over TLS, TLS binds the server certificate to domain or realm with ASN.1 encoding type [21]	ASN.1 is embedded into LDAP request and Response to communicate securely over TLS for authentication [20][20]
Vulnerability reported	No recursion limit or missing of end condition in BER for certificate generation in Kerberos [22]	In Samba 4.x versions validating ASN.1 memory allocation in LDAP server [23]
Possible impact	Denial of service attack for OpenStack Cloud [15]	Denial of service attack for file sharing server [23]

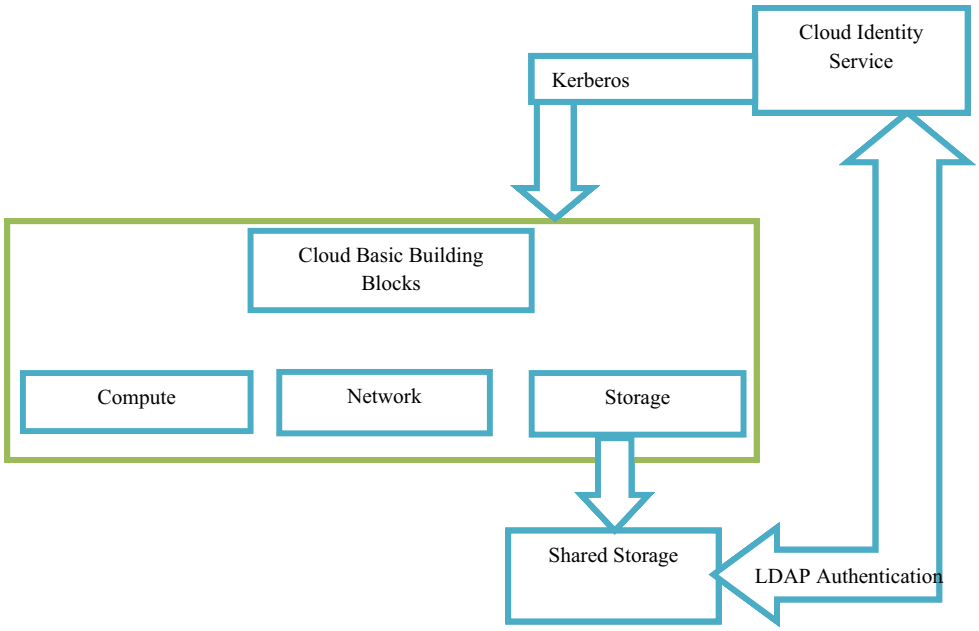

Fig. 3 Infrastructure model of Kerberos and LDAP for OpenStack

to enable encoding during transmission over the network to validate the identity and thereby communication. The critical vulnerability Denial of Service attack is reported due to infinite loop or memory allocation in ASN.1 implementation in Kerberos and LDAP authentication protocols, respectively. These vulnerabilities may have severe impact while authenticating process when extended to cloud or shared file storage. As industry and educational institutes are moving towards the cloud deployment to access the infrastructure over the network, these vulnerabilities may result in larger potential impact while working online in continues mode.

As a future work, the proposed model portrayed in Fig. 3 can be implemented and tested for a small user base initially, which can be further extended to support large user base to access cloud. This will help to interpret the possible attack scenario in Kerberos and LDAP while deploying with cloud and shared storage.

Declarations

Conflict of Interest The authors declare that they have no conflict of interest.

References

1. Management Information Base. Retrieved from Wikipedia, the free encyclopedia: https://en.wikipedia.org/wiki/Management_information_base (2021).
2. ASN.1. Retrieved from Wikipedia, the free encyclopedia: https://en.wikipedia.org/wiki/ASN.1 (2021).
3. Dubuisson O: ASN.1 communication between heterogeneous systems; 2000.
4. Kurose JF, Ross KW. Computer networking : a top-down approach. London: Pearson Education; 2017.
5. Kaliski Jr BS. A layman's guide to a subset of ASN.1, BER, and DER. RSH Data Security Inc., Redwood City (1993).
6. Microsystems, S. (n.d.). Basic encoding rules. Retrieved from Sun Microsystems: https://docs.oracle.com/cd/E19476-01/821-0510/def-basic-encoding-rules.html.
7. X.690. Retrieved from Wikipedia, the free encyclopedia: https://en.wikipedia.org/wiki/X.690 (2021).
8. Housley R. rfc5878. Retrieved from ietf.org: https://tools.ietf.org/html/rfc5878 (2010).
9. Whelan E. SNMP and potential ASN.1 vulnerabilities. Retrieved from SANS: https://www.sans.org/reading-room/whitepapers/protocols/paper/912 (2003).
10. Steiner JG, Neuman C, Schiller JI. Kerberos: an authentication service for open network systems. Retrieved from https://www3.nd.edu/~dthain/courses/cse66771/summer2014/papers/kerberos.pdf (1988).
11. Wang C, Feng C. Security analysis and improvement for kerberos based on dynamic password and Diffie-Hellman algorithm. In: Fourth international conference on emerging intelligent data and web technologies. China: IEEE; 2013. p. 256–60.
12. Kohl JT, Neuman BC, Ts'o TY. The evolution of the kerberos authentication service. In: Brazier F, Johansen D, editors. Distributed open systems. New Jersey: IEEE Computer Society Press; 1994. p. 78–95.
13. Bellovin SM, Merritt M, AT&T Bell Labs. Limitations of the Kerberos authentication system. [Online] https://people.eecs.berkeley.edu/~fox/summaries/glomop/kerb_limit.html.
14. Medvinsky A. rfc2712. Retrieved from ietf: https://tools.ietf.org/html/rfc2712 (1999).
15. SUSE SLES12 Security Update : krb5 (SUSE-SU-2020:3379–1). Tenable. [Online] 11 19, 2020.
16. Koutsonikola V, Vakali A. LDAP: framework, practices, and trends. IEEE Internet Comput. 2004;8(5):66–72.

17. System, Cisco. PKI data formats. https://www.cisco.com/. [Online] https://www.cisco.com/c/en/us/support/docs/security/vpn-client/116039-pki-data-formats-00.pdf.
18. Lim SS, Choi JH, Zeilenga KD. Design and implementation of LDAP component matching for flexible and secure certificate access in PKI. In Proc. of the 4th Annual PKI R&D Workshop; 2005, p. 41–51.
19. Sermersheim JE. rfc4511#section-4.1.11. Retrieved from ietf: https://tools.ietf.org/html/rfc4511#section-4.1.11 (2006).
20. Hodges J, Morgan R. rfc2830. Retrieved from ietf: https://tools.ietf.org/html/rfc2830 (2000).
21. Josefsson S. Using Kerberos version 5 over the transport layer security (TLS) protocol. Request for Comments (RFC), 6251 (2011).
22. CVE-2020–28196. Retrieved from CVE: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-28196 (2020).
23. CVE-2020–10704. Retrieved from Samba: https://www.samba.org/samba/security/CVE-2020-10704 (2010).
24. McCormick J. Serious Kerberos flaws affect Cisco and Mac, but not Windows. techrepublic.com. [Online] 9 14, 2004.
25. O'Reilly. (n.d.). LDAP. Retrieved from O'Reilly home.
26. Masud S. Kerberos-based authentication for OpenStack cloud infrastructure as a service. Texas: The University of Texas at San Antonio; 2014.