Abstract

Authentication is a fundamental aspect of system security. It confirms the identity of any user trying to log on to a domain or access network resources. Windows Server 2003 family authentication enables single sign-on to all network resources. With single sign-on, a user can log on to the domain once, using a single password or smart card, and authenticate to any computer in the domain. When attempting to authenticate a user, several industry-standard types of authentication may be used, depending on a variety of factors. We have different authentication protocols i.e. Kerberos, SSL/TLS, NTML, digest authentication etc. Basic purpose of this paper is focused on presenting a study between Kerberos and NTML.

1. Introduction

Growth and widespread use of computer networks and particularly the Internet in many activities of organizations and institutions makes widespread changes in human's lifestyles. Information security is an essential and important issue in this area. Connecting local network of an organization to global networks places data of organizations in risk of exposed access by external hosts. Confidentiality of sensitive information from unauthorized access on the internet is the most important security challenge. Authentication is one of the basic mechanisms for enforcing confidentiality. Authentication [6] is process of identifying and ensuring for identity of a party for enforcing confidentiality preventing impersonation. Authentication is one of the key fields of security in the information exchange network. Authentication verifies the identity of every user who wants to access network's resources.

When computers were stand-alone entities, security was provided by physical means: computer rooms were sealed and locked, and punched cards went in one window with line printer listings out another. But since the early 1970s, networking brings up the necessity to communicate securely over an insecure network of computers, where multiple users can gain access to the network services. Therefore, the ability to accurately identify each user making a request becomes essential.
In general terms, the problem is how the provider of a service can determine whether a client's request for the service is to be honoured or not. A good solution is the server's capability to verify the user's identity. This process is called authentication and goes through some steps which constitute what is called an authentication protocol. Although protocols are generally made up of a few messages sent on the network, they can hide terribly subtle errors. It is not unusual for errors to be discovered on protocols that have been used for years. Protocol errors are unlikely to be detected in normal operations. The need for techniques to verify the correctness of such protocols is great [7]. A protocol should be correct, i.e. it should allow authorised users from gaining the services they require, and hopefully secure, i.e. it should prevent any unauthorised user to get access to any service. Unfortunately, unlike the former notion, security is not a simple Boolean predicate” [5]. After thirty years of networking, researchers have proved many properties of different protocols, but claims like ‘this protocol is secure’ are still difficult to make. The two commonly used protocols are NTLM and Kerberos.

2 Overview of NTLM
NTLM family protocols are embedded in applications the application caller sends parameters to the NTLM which then reply with an authentication method that the caller uses into its own messages which are to be transmitted.

- **NEGOTIATE_MSG (Client to server):** To begin the communication client sends the message (Request) to the server to access the service. In this message the client specifies its supported NTLM options to the server. The client shows its capabilities to the server whether it can use NTLM family protocols or not. This message can contain different fields like domain names, work station name version structure and payload (variable).

- **CHALLENGE_MSG (Server to Client):** This message is send by the server to the client to challenge the client for proving its identity. This message is generated by server in response to the client’s negotiate message. This message can contain the fields like message type, target name, server challenge (8 bytes), target information, version structure and payload.

- **AUTHENTICATE _MSG (Client to Server):** This message is sent from client to server in response to the challenge given by the server. This contains fields like message type, challenge
response (8 bytes), domain name, user name, work station, encrypted random session key, version structure, message integrity and payload.

2.1 NTLM v1
It works on the basic model of NTLM family explained above. The server and the client share a secret key. When the communication starts, the client requests the server through a negotiate message. The server sends an 8 byte random number as challenge (CH) to the client in response to its negotiate message (NEG_MSG). After receiving the 8 bytes random number the client compute the value using challenge and secret key (one of the two password hashes). This computed value returned by the client to the server as 24 bytes result (Response1, Response2). The server now verifies the client’s identity using its response (client_response). The server checks whether the value commuted by the client is correct or not. If the value is correct then the client is valid user and hence authenticated.

In NTLM v1 [6] client uses both hashes (NT-hash and LM-hash) to compute the value and send both results to the server in 24 bytes packet. The hashes produce 16 bytes quantities. To make these quantities 21 bytes, five bytes of zeros are added at the end of 16 bytes quantities. The 21 bytes are then divided into three seven bytes (56 bits) values (key1, key2, key3). These 56 bits values used as secret keys for encryption by DES. By using these 64 bytes, challenge is encrypted and sent to the server.

\[
\text{CH} = 8\text{bytes server challenge (randomnumber)} \\
\text{key1/key2/key3} = NT - \text{hash} - 5\text{byte} - 0 \\
\text{Response1} = \text{DES(key1, CH)}/\text{DES(key2, CH)/DES(key3, CH)} \\
\text{key1/key2/key3} = LM - \text{hash} - 5\text{byte} - 0 \\
\text{Response2} = \text{DES(key1, CH)/DES(key2, CH)/DES(key3, CH)} \\
\text{client_response} = \text{Response1/Response2}
\]

2.2 NTLM v2
The working of NTLM v2 is slightly different from NTLM v1. It sends two 16 bytes responses (client_response (LMv2 and NTv2)) to an 8 bytes server challenge (SCH (random number)). Instead of DES it uses HMAC-MD5 algorithm to compute the value on the client machine. The first response (LMv2) contains HMAC-MD5 hash of server challenge (SCH), a client challenge (CCH (random number)), client’s password hashed by HMAC-MD5 (v2-hash) and other client related information (identity information). The second response (NTv2) sent by NTLM v2 from client machine uses client challenge of variable length (CCH*). The client challenge (CCH*) contains a) the current time, b) 8 bytes random value (CCH), c) domain name and d) standard format values (MSG_format_values).

\[
\text{SCH} = 8\text{bytes server challenge (randomnumber)} \\
\text{CCH} = 8\text{bytes client challenge (randomnumber)} \\
\text{CCH *} = (\text{MSG_format_values, current_time, CCH, domain_name}) \\
\text{v2} = \text{hash} = \text{HMAC - MD5(NT - hash, client_name, domain_name)} \\
\text{LMv2} = \text{HMAC - MD5(v2 - hash, SCH, CCH)} \\
\text{NTv2} = \text{HMAC - MD5(v2 - hash, SCH, CCH *)} \\
\text{client_response} = \text{LMv2/CCH/NTv2/CCH *}
\]

2.3 Kerberos
The Kerberos in Greek Mythology means a three headed dog which stands at the gates of house of Hades which let the dead passes by to enter and eat anyone try to go back to the land of living. Here the three heads denotes the AAA known as Authentication, Authorization and Accounting. These three are the basic requirements for designing any secure environment.

The Kerberos protocol is designed & developed to provide authentication services. The MIT designed Kerberos for providing authentication across unsecure networks by using the private key cryptography. The authentication in Kerberos is done through a trusted third party who is denoted as an authentication server. Till now MIT developed five versions of Kerberos. Version 1 through 3 was used internally by MIT version 4 is accepted beyond MIT. Models for administration and use of computer services differ from site to site and some environments require support that isn’t present in Version 4. Version 5 of the Kerberos protocol incorporates new features suggested by experience with Version 4, making it useful in more situations. Version 5 is based in part upon input from many contributors familiar with Version 4.

"Kerberos is an authentication protocol for trusted hosts on untrusted networks". Kerberos is not beneficial if the host is not trustworthy. Otherwise, the intruder can use host as a key to get authentic. Intruder can impersonate by obtaining IP address for that server.

When using authentication based on cryptography, an attacker listening to the network gains no information that would enable it to falsely claim another's identity. Kerberos is the most commonly used example of this type of authentication technology.

Modern computer systems provide service to multiple users and require the ability to accurately identify the user making a request. In traditional systems, the user's identity is verified by checking a password typed during login; the system records the identity and uses it to determine what operations may be performed. The process of verifying the user's identity is called authentication. Password based authentication is not suitable for use on computer networks. Passwords sent across the network can be intercepted and subsequently used by eavesdroppers to impersonate the user. While this vulnerability has been long known, it was recently demonstrated on a major scale with the discovery of planted password collecting programs at critical points on the Internet.

3. Need of study

Massachusetts Institute of Technology (MIT) developed Kerberos to protect network services provided by Project Athena. Several versions of the protocol exist; versions 1-3 occurred only internally at MIT. Many members of Project Athena contributed to the design and implementation of Kerberos. In there is a dialogue that was written in 1988 to help its readers understand the fundamental reasons for why the Kerberos V4 protocol was the way it was. It was amazing how much this dialogue was still applicable for the Kerberos V5 protocol. Although many things were changed, the basic core ideas of the protocol have remained the same. Miller and Neuman are the primary designers of Kerberos Version 4 with contributions from Saltzer and Schiller. They published that version in the late 1980s, although they had targeted it primarily for Project Athena. Version 5, designed by Kohl and Neuman, appeared as RFC 1510 in 1993 (made obsolete by RFC 4120 in 2005), with the intention of overcoming the limitations and security problems of version 4. The Kerberos authentication system is based on the trusted 3rd party Needham-Schroeder authentication protocol. The system is one of the few industry standards for authentication systems and its use is becoming fairly widespread. The system has some limitations, including the fact that compromise of the on-line trusted 3rd party is catastrophic and that the system is vulnerable to dictionary attacks. Further, while the system provides for authentication and key-exchange, it does
not provide non repudiation (i.e. digital signature) services, as a result of which an organization using Kerberos would have to maintain a separate security infrastructure for the latter function.

4. Comparative analysis

The following table presents the comparison of NTLM and Kerberos:

Table 1: Comparison of NTLM and Kerberos

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Cryptographic Technique</th>
<th>Security Level</th>
<th>Message Type</th>
<th>Ease to Use</th>
<th>Trusted Third Party</th>
<th>Microsoft Supported Platforms</th>
<th>Client / Server Challenge</th>
</tr>
</thead>
<tbody>
<tr>
<td>NTLM v1</td>
<td>Symmetric Key Cryptography</td>
<td>Less</td>
<td>Random number</td>
<td>Easy</td>
<td>Domain Controller</td>
<td>Win 95, 98, ME, NT 4.0, 2000, XP, 2003/R2, Vista</td>
<td>Server Only</td>
</tr>
<tr>
<td>NTLM v2</td>
<td>Symmetric Key Cryptography</td>
<td>More Than NTLM v1</td>
<td>MD4 hash &amp; Random number</td>
<td>Easy</td>
<td>Domain Controller</td>
<td>Win 95, 98, ME, NT 4.0, 2000, XP, 2003/R2, Vista</td>
<td>Both Client &amp; server</td>
</tr>
<tr>
<td>Kerberos v4</td>
<td>Symmetric Key Cryptography and Asymmetric Cryptography</td>
<td>More than NTLM v1 and v2</td>
<td>Encrypted Ticket Using DES</td>
<td>Complex</td>
<td>Domain Controller &amp; Key Distribution Center</td>
<td>Win2K, XP, Windows 2003/R2, Vista</td>
<td>Both Client &amp; server</td>
</tr>
<tr>
<td>Kerberos v5</td>
<td>Symmetric Key Cryptography and Asymmetric Cryptography</td>
<td>More than Kerberos v4</td>
<td>Encrypted Ticket Using MD5</td>
<td>Complex</td>
<td>Domain Controller &amp; Key Distribution Center</td>
<td>Win2K, XP, Windows 2003/R2, Vista, 7, 8</td>
<td>Both Client &amp; server</td>
</tr>
</tbody>
</table>

5. Conclusion

There are many other authentication protocols also used. In this we have presented only the best known. NTML and Kerberos have been discussed in this paper briefly. In general we can say that, NTML uses challenges for authentication whereas in Kerberos time synchronisation is used which provides higher security level. Although Kerberos is still in use.

References

Tanuja Thakur: Comparative Analysis Of Window Authentication Protocols


