Attacking NTLMv1/NTLMv2 with SSPI-NTLMSSP Message Protocol

George Fekkas

g [dot] fekkas [at] encodegroup [dot] com

"The highest technique is to have no technique. My technique is a result of your technique; my movement is a result of your movement."

Bruce Lee

Last Modified: 10/5/2014

Contents

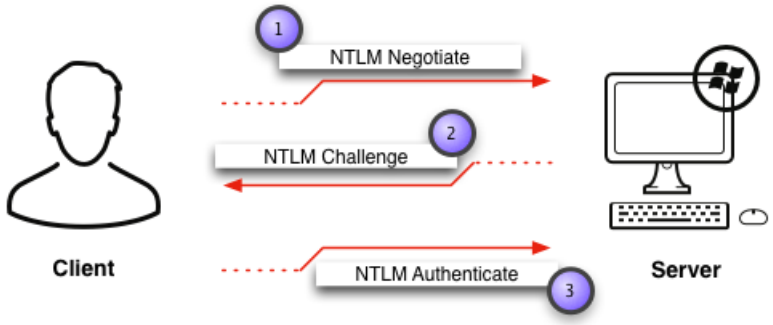
1 Intorduction

During a Cyber Threat Assessment or an Advanced Persistent Threat Readiness exercise an initial foothold is established to the internal network through the end user's workstation. If the current logged on user is a local administrator then the penetration tester is able to retrieve system hashes from the memory using tools such as mimikatz, wce, pwdumpX e.t.c. Let's imagine that an internal user's workstation is compromised, however normal user access has been granted (e.g. not as a local administrator). How the penetration tester will retrieve the password of the current logged on user? Social engineering is an potential option, where the penetration tester can upload a malicious application which will display a fake pop-up login (e.g. e-mail client) in the user's screen. Then, potentially, the end user will fill the text boxes, username and password respectively. The malware will create a text file with the user's credentials. In this paper we describe a different technique where a penetration tester is able to capture the hash of the current logged on user (even if the user has not local administrator privileges) through the Window Security Support Provider Interface (SSPI) framework.

2 The NTLM Challenge

NTLM authentication is a challenge-response scheme, consisting of three messages, commonly referred to as Type 1 (negotiation), Type 2 (challenge) and Type 3 (authentication). It basically works like this:

- The client sends a Type 1 message to the server. This primarily contains a list of features/options supported by the client and requested by the server.
- The server responds with a Type 2 message. This contains a list of features supported and agreed upon by the server. Most importantly, however, it contains an (8 byte) challenge (Nonce) generated by the server.
- The client replies to the challenge with a Type 3 message. This contains several pieces of information about the client, including the domain and username of the client user. It also contains one or more responses to the Type 2 challenge. **The response in the Type 3 message is the most critical piece, as it proves to the server that the client has knowledge of the account password.** [1]



The **Type 1 message** is sent from the client to the server to initiate the NTLM authentication. Its primary purpose is to establish the "ground rules" for authentication by indicating supported options via the flags. Optionally, it can also provide the server with the client's workstation name and the domain in which the client workstation has membership; this information is used by the server to determine whether the client is eligible for local authentication. [1] An example of a Type Message 1 (NTLM Negotiate), is as follows:

```
00000000: 4E 54 4C 4D 53 53 50 00  01 00 00 00 B7 82 08 E2  NTLMSSP.........
00000010: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000020: 05 01 28 0A 00 00 00 0F                          ..(.....
```

The **Type 2 message** is sent by the server to the client in response to the client's Type 1 message. It serves to complete the negotiation of options with the client, and also provides a challenge to the client. It may optionally contain information about the authentication target.[1] An example of a Type Message 2 (NTLM Challenge), is as follows:

```
00000000: 4E 54 4C 4D 53 53 50 00  02 00 00 00 10 00 10 00  NTLMSSP.........
00000010: 38 00 00 00 35 82 8A E2  6A 26 44 3E BB 19 CB 7E  8...5...j&D>...~
00000020: 00 00 00 00 00 00 00 00  54 00 54 00 48 00 00 00  ........T.T.H...
00000030: 05 01 28 0A 00 00 00 0F  4E 00 45 00 54 00 53 00  ..(.....N.E.T.S.
00000040: 50 00 41 00 52 00 4B 00  02 00 10 00 4E 00 45 00  P.A.R.K.....N.E.
00000050: 54 00 53 00 50 00 41 00  52 00 4B 00 01 00 10 00  T.S.P.A.R.K.....
00000060: 4E 00 45 00 54 00 53 00  50 00 41 00 52 00 4B 00  N.E.T.S.P.A.R.K.
00000070: 04 00 10 00 6E 00 65 00  74 00 73 00 70 00 61 00  ....n.e.t.s.p.a.
00000080: 72 00 6B 00 03 00 10 00  6E 00 65 00 74 00 73 00  r.k.....n.e.t.s.
00000090: 70 00 61 00 72 00 6B 00  00 00 00 00              p.a.r.k.....
```

The Server Challenge (or Nonce) in the above example is the 8-byte **6A26443EBB19CB7E**.

The **Type 3 message** is the final step in authentication. This message contains the client's responses to the Type 2 challenge, which demonstrate that the client has knowledge of the account password without sending the password directly. The Type 3 message also indicates the authentication target (domain or server name) and username of the authenticating account, as well as the client workstation name.[1] An example of a Type Message 3 (NTLM Authenticate), is as follows:

```
00000000: 4E 54 4C 4D 53 53 50 00  03 00 00 00 18 00 18 00  NTLMSSP.........
00000010: 70 00 00 00 84 00 84 00  88 00 00 00 10 00 10 00  p...............
00000020: 48 00 00 00 08 00 08 00  58 00 00 00 10 00 10 00  H.......X.......
00000030: 60 00 00 00 10 00 10 00  0C 01 00 00 35 82 88 E2  `...........5...
00000040: 05 01 28 0A 00 00 00 0F  4E 00 45 00 54 00 53 00  ..(.....N.E.T.S.
00000050: 50 00 41 00 52 00 4B 00  53 00 53 00 50 00 49 00  P.A.R.K.S.S.P.I.
00000060: 4E 00 45 00 54 00 53 00  50 00 41 00 52 00 4B 00  N.E.T.S.P.A.R.K.
00000070: 77 4F FE 79 FB 92 37 FA  71 8B 85 55 F6 99 FB BE  wO.y..7.q..U....
00000080: 73 A0 C2 CE 2E 99 D3 A1  65 B9 10 5D DA AC 2D 94  s.......e..]..-.
00000090: 69 D0 58 2E 9C 3A 06 32  01 01 00 00 00 00 00 00  i.X..:.2........
000000A0: D8 E2 78 12 40 62 CF 01  73 A0 C2 CE 2E 99 D3 A1  ..x.@b..s.......
000000B0: 00 00 00 00 02 00 10 00  4E 00 45 00 54 00 53 00  ........N.E.T.S.
000000C0: 50 00 41 00 52 00 4B 00  01 00 10 00 4E 00 45 00  P.A.R.K.....N.E.
000000D0: 54 00 53 00 50 00 41 00  52 00 4B 00 04 00 10 00  T.S.P.A.R.K.....
000000E0: 6E 00 65 00 74 00 73 00  70 00 61 00 72 00 6B 00  n.e.t.s.p.a.r.k.
000000F0: 03 00 10 00 6E 00 65 00  74 00 73 00 70 00 61 00  ....n.e.t.s.p.a.
00000100: 72 00 6B 00 00 00 00 00  00 00 00 00 81 8F 1A F0  r.k.............
00000110: CD F1 5E EB 73 F6 5E 6A  D3 9D 50 EF              ..^.s.^j..P.
```

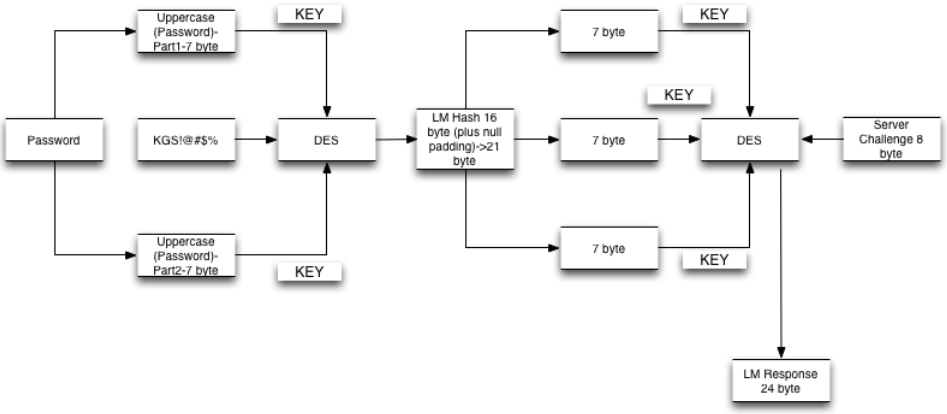
The NTHash in the above example is the 16-byte **65b9105ddaac2d9469d0582e9c3a0632**. The Client Challenge (variable) is the **0101000000000000...0000000000000000**.

3 LM/NTLM-LMv2/NTLMv2

Microsoft Windows supports two primary algorithms for authenticating users. These algorithms generate what's known as an LM Hash and NT Hash respectively.

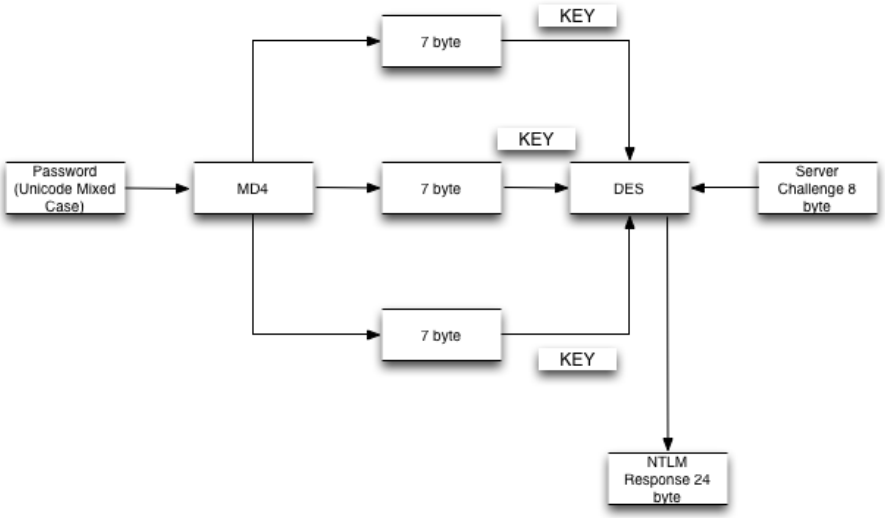
3.1 LM Response

LM hash is enabled by default in Windows NT, Windows 2000, Windows XP, Windows 2003. In the below figure we can see how the LM Hash (24 byte) is generated. The below scheme is older than the NTLM response, and less secure. While newer clients support the NTLM response, they typically send both responses for compatibility with legacy servers.



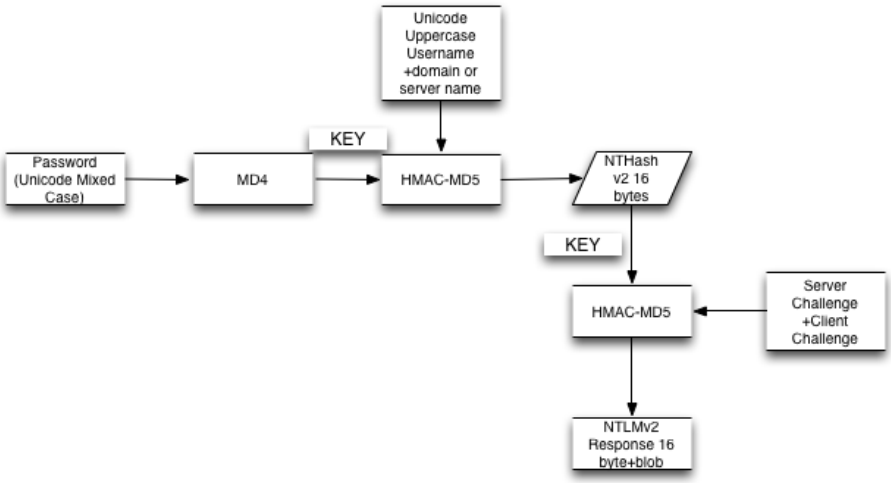
3.2 NTLM Response

In the figure below we can see how the NTLM Hash 24 byte is generated. The NTLM Response addresses has security related flaws in the LM response. The NTLM response is always sent with the LM response.



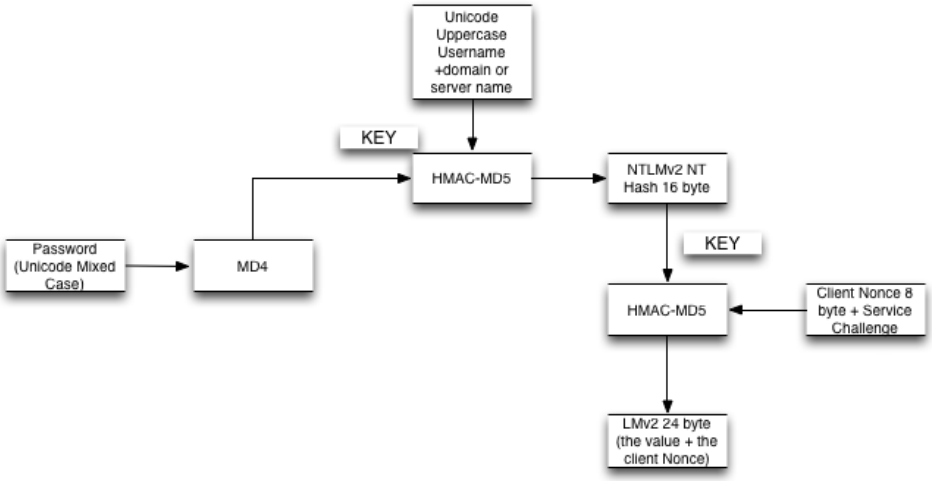
3.3 NTLMv2 Response

NTLM version 2 ("NTLMv2") was developed, as an evolution, to address the security issues present in the NTLM protocol. When NTLMv2 is enabled, the NTLM response is replaced with the NTLMv2 response. The NTLMv2 response is sent by newer clients such as Windows Vista, 7, Windows 8, Windows Server 2008/2012. The NTLMv2 response protocol employees a unique client challenge, same as the LMv2 response protocol (Section 3.4). This additional data effectively defeats the ability to precompute password/response pairs via Rainbow Tables. In the figure below we can see how the NTLMv2 Hash 16 byte is generated.



3.4 LMv2 Response

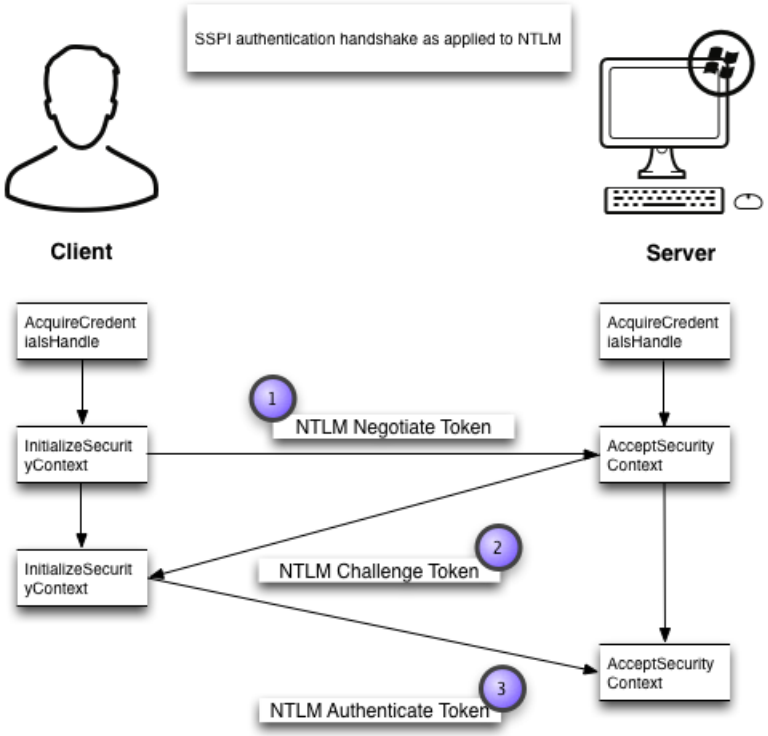
The LMv2 response protocol employees a unique client challenge as well, same as NTLMv2 protocol. This additional data effectively defeats the ability to precompute password/response pairs via Rainbow Tables.The LMv2 response protocol is used to provide pass through authentication with older servers, for compatibility reasons. In the figure below we can see how the LMv2 Hash 24 byte is generated.



For more information related to the LMv1/NTLMv1 & LMv2/NTLMv2 see bibliography-References [1][2][4]

4 NTLMSSP-SSPI (Local Authentication)

The NTLMSSP [5] (NT LAN Manager Security Support Provider) is a messaging protocol, a binary one, used by the Security Support Provider Interface (SSPI). The purpsose of this protocol is to facilitate NTLM challenge-response authentication between a Client and Server, as we saw in the previous section "2. LM/NTLM-LMv2/NTLMv2". Windows provides a security framework known as SSPI - the Security Support Provider Interface. SSPI is a well-defined API for obtaining integrated security services for, among other things, authentication for any distributed application protocol. This is the Microsoft equivalent of the GSS-API (Generic Security Service Application Program Interface, RFC 2743), and allows for a very high-level, mechanism-independent means of applying authentication, integrity, and confidentiality primitives.SSPI supports several underlying providers; one of these is the NTLMSSP (NTLM Security Support Provider), which provides the NTLM authentication mechanism. SSPI supplies a flexible API for handing authentication tokens such as the NTLM Type 1 (Negotiate), Type 2 (Challenge), and Type 3 (Response) messages. These token messages are processed by the NTLMSSP messaging protocol as follows:



.

The following steps are applied (SSPI authentication handshake-NTLM):

1. The client obtains a representation of the credential (Username, Domainname) set for the user via the SSPI AcquireCredentialsHandle function.

2. The client calls the SSPI InitializeSecurityContext function to obtain an authentication request token (in our case, a Type 1 message). The client sends this token to the server. The return value from the function indicates that authentication will require multiple steps.

3. The server receives the token from the client, and uses it as input to the AcceptSecurityContext SSPI function. This creates a local security context on the server to represent the client, and yields an authentication response token (the Type 2 message), which is sent to the client. The return value from the function indicates that further information is needed from the client.

4. The client receives the response token from the server and calls InitializeSecurityContext again, passing the server's token as input. This provides us with another authentication request token (the Type 3 message).

5. The return value indicates that the security context was successfully initialized; the token is sent to the server. The server receives the token

from the client and calls AcceptSecurityContext again, using the Type 3 message as input. The return value indicates the context was successfully accepted; no token is produced, and authentication is complete.

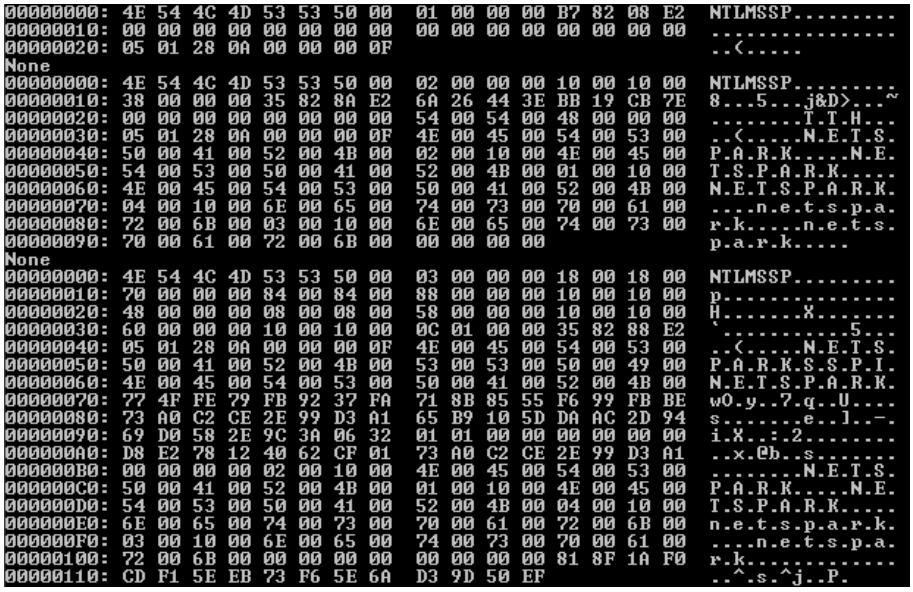
For more information related to the SSPI authentication see bibliography-References [3]

5 Tool-The Magic Key

The **Magic Key** is a tool, (Python language), that is able to retrieve the hash of the current logged on user using the SSPI security framework through the NTLMSSP message protocol. By using this method we are able to retrieve the NTLMv1/NTLMv2 hash even if the user has no administrative privileges, in our case we are using the user "SSPI". The user "SSPI" does not belong to the local administrator group. The tool passes the win32api.GetUserName(), win32api.GetDomainName() and the password = None to the auth_info parameter to the AcquireCredentialsHandle function which is responsible to create a handle for the SSPI context . Then the Client calls the SSPI InitializeSecurityContext function to obtain an authentication request token (Type 1 Message). The three-way dance authentication is occured as described in the section "3. NTLMSSP-SSPI (Local Authentication)". Then the tool examines the Type 2 Message and Type 3 Message respectively in order to retrieve the server challenge (Type 2 message) and the LM/NTLM Hashes (24 bytes each) in a case of NTLMv1 and the NTHash (16 bytes) plus the ClientChallenge (variable) in a case of NTLMv2. The detection of the NTLMv2 is done through the "magic" string " 0101000000000000". Finally, the tool constructs the hash format for NTLMv1/NTLMv2 in a such way, which it is ready to pass it to various cracking tools such as John the Ripper [5] and Hashcat [6] as well.

The -v option can be used in order to display the dance authentication process (Type1/Type2/Type3) as follows:

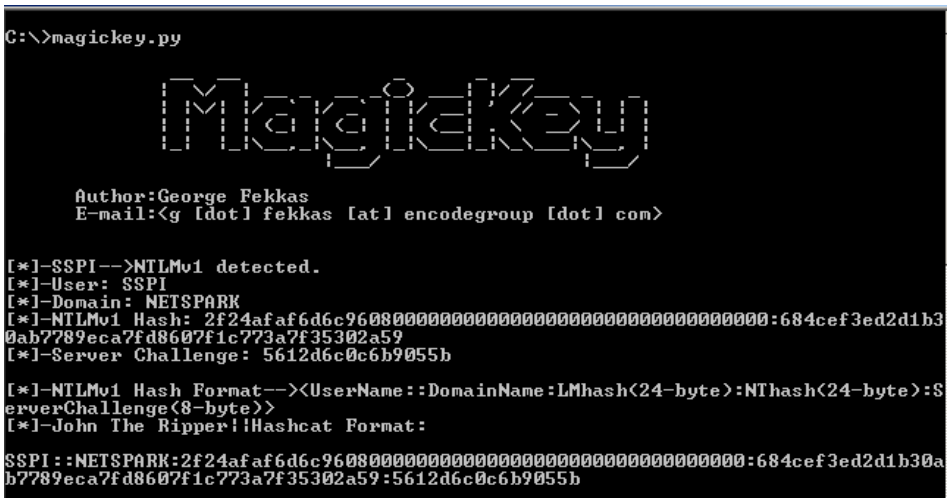
- **Type 1:** NTLM Negotiate Message
- **Type 2:** NTLM Challenge Message
- **Type 3:** NTLM Authenticate Message



5.1 NTLMv1 Output

Network security: LAN Manager authentication level

- **Send LM & NTLM Responses**
- **Send LM & NTLM Responses-use NTLMv2 session security if negotiated**
- **Send NTLM response only**



the following hash format can be retrieved and cracked offline:

- **Username::DomainName:LMHash(24byte):NTHash(24byte):ServerChallenge(8Byte)**

5.2 NTLMv2 Output

Network security: LAN Manager authentication level

- **Send NTLMv2 Response only**
- **Send NTLMv2 Response only/refuse LM**
- **Send NTLMv2 Response only/refuse LM &NTLM**

the following hash format can be retrieved and cracked offline:

- **Username::DomainName:ServerChallenge(8byte):NTHah(16byte):ClientChallenge(variable)**

6 Bibliography

References

[1] The NTLM Authentication Protocol and Security Support Provider-http://davenport.sourceforge.net/ntlm.html

[2] [MS-NLMP]: NT LAN Manager (NTLM) Authentication Protocol-http://msdn.microsoft.com/en-us/library/cc236621.aspx

[3] SSPI Authentication http://msdn.microsoft.com/en-us/library/aa380493(v=vs.85).aspx

[4] NT LAN Manager http://en.wikipedia.org/wiki/NT_LAN_Manager

[5] NTLMSSP-https://en.wikipedia.org/wiki/NTLMSSP

[6] John the ripper-http://www.openwall.com/john/

[7] Hashcat-http://hashcat.net/oclhashcat/

7 Acronyms

DES: Data Encryption Standard
HMAC: Hash Message Authentication Code
GSS-API: Generic Security Service Application Program Interface
LM: Lan Manager
MD: Message Digest
NTLM: NT LaN Manager
NTLMSSP: NT Lan Manager Security Support Provider
SSPI: Security Support Provider Interface