

NTLM Authentication Unsafe

Originally presented at
Symposium on Security for Asia Network
SyScAN '04

December 17th, 2004 – some updates for March 2005



Information Security Partners, LLC
iSECPartners.com



What you get for attending – An Outline

- **Demo of an easy-to-execute attack on NTLM**
 - Works on majority of corporate networks.
- **How to harden against attack.**
- **Enumeration of NTLM's risks, so that you can justify the deployment of potentially inconvenient settings and have a clear understanding of the risks you must accept.**
- **I will talk about other weaknesses of NTLM, including password authenticators being password equivalents, what that means, and demonstrate an iSEC tool for taking advantage of it.**
- **I will discuss authentication alternatives including SRP (a zero-knowledge proof of knowledge protocol)**

Agenda

- **Background on NTLM**
 - What is NTLM
 - A little crypto
 - Some observations
- **Password equivalence issues**
- **Pre-computed dictionary attacks**
 - NTLMv2 as a solution
- **Middle person attack**
- **Hardening against these issues**
- **Questions**

What is NTLM?

- **An authentication protocol that supports Single Sign-On primarily used on Windows networks**
- **Challenge - Response based**
- **Replaced a weaker protocol called “LAN Manager”**
- **NTLM is “NT Lan Manager”**
- **NTLM is NTLM version 1 or NTLMv1**
- **Still “on” by default in popular operating systems**
- **Ubiquitous – Windows, Mac and Linux networks all commonly deploy NTLM**
 - Windows XP / Server 2003 / 2000 / NT
 - Internet Explorer
 - Firefox
 - Samba
 - jCIFS
 - Mac OS X
 - WinCE
 - Storage devices

What is NTLM?

- **Used to authenticate**
 - Web site users – share point, IIS, servlets with jCIFS
 - File share access – including default shares like c\$ and admin\$
 - Printer access
 - DCE RPC calls
 - DCOM
- **Usually can be replaced with stronger Kerberos authentication**
 - But active attackers can downgrade to NTLM
 - Users without tickets, or with intermittent connectivity avoid Kerberos
- **Dictionary attacks have been demonstrated against NTLM**
 - Cain and Abel, l0pht crack et al. perform password search against sniffed NTLM authentications
 - These are not pre-computed attacks – so they require very weak passwords

NTLM Basics

1. Take a password
2. Calculate the MD4 of it -> 128 random bits or 16 bytes
3. Store this value – the “NTLM Hash” (aka the Authenticator) – this is different than the “LAN Man Hash”

During Authentication

1. Get an 8 byte challenge
2. Divide the NTLM Hash (MD4 of password) into three pieces
 - 2 seven byte and one 2 byte ($7 + 7 + 2 = 16$ bytes)
3. DES encrypt the challenge with the three MD4 chunks
4. Concatenate the three cipher texts to make the “NTLM response”
5. Send the “NTLM response” to the server

NTLM Basics

How an NTLM Hash & Challenge → NTLMv1 Response

MD4 of password = 0x**0123456789ABCDEF****FFEDCBA9876543210**

→ Broken into three “Key Chunks”

→ Key 1: **0123456789ABCD**

→ Key 2: **FFEDCBA987654**

→ Key 3: **32100000000000**

NTLM Response is the concatenation of the encryption of the challenge with three separate keys =

DES(Challenge, **Key1**) + DES(Challenge, **Key2**) + DES(Challenge, **Key 3**)

NTLM Basics

- Response generation requires only the “NTLM Hash” (which is the password authenticator). The server must store the password authenticator to be able to authenticate users.
- Unix password authenticators typically are non-password-equivalent salted hashes so that exposure isn't catastrophic.
- Effort to try all DES keys: $2 \times 2^{56} + 2^{16} \cong 2^{57}$ for all three portions of the response
- 10 character alphabetic mixed case password has about 2^{57} possibilities

NTLM Observations – What's nice

- NTLM provides a different response for every challenge, so an attacker can't pre-compute a dictionary attack as the challenge space is huge (2^{128} is bigger than my golf score)
- Doesn't require a centralized server available at logon
- Can "pass through" authorization to a domain controller or other server
- Passive attackers don't see passwords
- No replay attacks as the server is unlikely to ever reuse a challenge (about 2^{64} observations required for a collision)
- MD4 of the password whitens it

NTLM Observations – What's weak

- The password authenticator is itself password equivalent
- The authenticator is protected with the weak DES cipher
- Machines can break DES – cost and speed are not public
- DES cipher has an attack complexity of 2^{56} , which means for [a-z][A-Z] passwords of length 10 or more it is fewer operations to break DES than to brute force the password.
- Passive adversaries get an offline dictionary attack on the password which could compromise weak passwords (Cain and Abel)
- The protocol doesn't protect against middle person attacks
- Active attackers can provide known challenges and launch pre-computed dictionary attacks
- Most of these would be fine if the communication was over a secure SSL or TLS tunnel
 - The password equivalency of the authenticator would not

Password Equivalence of Authenticator

- To demonstrate the password equivalence of the NTLM Hash (MD4(password)) I modified the jCIFS “SmbShell” example so that it would authenticate using username and NTLM Hash instead of password.
- This only works because of the “password equivalence” of the NTLM Hash which is used as the “password authenticator”.
- Attacker doesn't have to use rainbow tables or brute force
- Extracting hashes does require administrator access!
- Often one time brief physical access can result in password hash extraction

Password Equivalence of Authenticator

- **Syskey helps protect windows machines that store these authenticators (i.e. machines with passwords)**
- **Local accounts are stored on workstations like this laptop**
- **Local administrator accounts often have synchronized passwords in corporate environments – while convenient for administration, the password equivalence of these authenticators represent an enormous risk!**

Password Equivalence of Authenticator

- This equivalence is commonly misunderstood.
- Many people assume the password hashes stored by Windows are similar to the password hashes found in BSD or Linux shadow files. However shadow hashes are generally not password equivalents.
- The equivalence is mentioned in the excellent book “Network Security - Private Communication in a public world” Second Edition, Radia Perlman, Charlie Kaufman, Mike Speciner
“the fact that the server stores a hash of the user’s password rather than the actual password does not theoretically make the scheme more secure. A modified version of the client software could impersonate the user if it directly used the hash of the password rather than hashing the string the user types.” – Page 622, 24.7.1 Lan Manager and NTLM
- The modified client I present demonstrates the practical truth of the above theoretical statement.

NTLM Hash Authentication Shell Demonstration

- **java -jar SmbShell.jar**
- **This shell is similar to the example in jCIFS except that it has been modified to:**
 - Login without passwords
 - Support deletion of files
 - Support uploading of files (put)
 - Support downloading of files (get)

DEMO

Protecting against authenticator password equivalence

- **Use Syskey with a password to keep workstations safer from attackers who get them alone with a floppy or USB disk**
 - Laptops or at-risk workstations are particularly good candidates
- **Never synchronize local passwords between machines!**
 - Especially not local administrator passwords (where synchronization is common)
- **Use Secure Remote Password protocol (SRP <http://srp.stanford.edu/>) for incompatible applications or systems that need to authenticate users by password**
- **Acknowledge and accept the risk**

Open Questions

- **Do you think NTLMv2 or SMB Signing resolve this?**
 - I don't believe so. Although I am not demonstrating equivalence with NTLMv2 or SMB Signing required.
- **Does Kerberos help?**
 - I am not yet convinced, but at least it helps against these tools.

Dictionary attacks

Dictionary attacks come in three flavors

- **Online – almost useless against any but the weakest passwords**
 - Hard to prevent, account lockout periods are commonly used to mitigate this minor risk
 - Works well on passwords like “password”, “Jesse”, and “test”
- **Offline – useful against weak passwords**
 - Real problem for obsolete “LAN Manager” protocol, not common against NTLM
 - Cain and Abel supports this kind of dictionary attack through monitoring of NTLM Challenge-Response pairs
 - Works well on passwords like “p0tat0”, “Cat2”, and “giantrobot”
 - Possible with a DES cracking oracle (targets recovery of the NTLM Hash)
- **Pre-computed offline – Very effective against whole key spaces with Time-Space trade-offs as seen in Rainbow Crack**
 - Possible with an active attack on NTLM
 - Works well on passwords like “ydDKaEza”, “a39dAJ1z”, and “b@d00D!”

Pre-computed dictionary attacks on NTLM

- **By deciding on a standard “challenge” and computing the dictionary, key space, or crack table based on that challenge an attacker prepares to break a password**
- **The attacker then impersonates a server**
 - This can be very easy
 - IE will magically authenticate against a site with a Netbios name if directed there with an iframe, link, or 302.
 - Firefox supports transparent authentication after the initial auth – initial authentication prompts the user.
- **The attacker’s server uses the standard challenge that was used to generate the dictionary, key space or crack table.**

Pre-computed dictionary attacks on NTLM

- **NTLMv2 is specifically designed to eliminate the possibility of pre-computed dictionary attacks**
 - Uses client nonce in addition to server nonce / challenge in calculation of response
 - An additional fix in NTLMv2 reduces the risk of middle person attacks
 - Additional nonce changes the size of the response breaking many implementations
- **NTLMv2 is probably an acknowledgement of the pre-computed dictionary weakness in NTLM**
- **It has been mentioned before online – as a letter in Christopher R. Hertel’s online book “Implementing CIFS – The Common Internet File System” reads:**

“you talk about the "client challenge" a bit, but miss the point of it: the client nonce (as it should really more correctly be called) is there to prevent precomputed dictionary attacks by the server ” - Ronald Tschalär to Chris Hertel, in Implementing CIFS <http://ubiqx.org/cifs/SMB.html#SMB.8>

Pre-computed dictionary attacks on NTLM

- This servlet negotiates NTLM authentication against browsers
- Always serves the challenge:

```
byte[] ISEC_CHALL = { (byte) 'i', (byte) 'S',  
    (byte) 'E', (byte) 'C', 0, 0x15, (byte) 0xEC, 0  
    };
```

- In hex: 695345430015EC00
- Under NTLM each password maps to one NTLM response
- Under NTLMv2 each password maps to many NTLMv2 responses

DEMO

Protecting against pre-computed dictionary attacks

- Require only NTLMv2 or LMv2 on all servers and workstations
- Negotiating NTLMv2 does nothing to protect you, the active attack requires an active adversary and active adversaries can easily downgrade.
- Change the settings from default – this ships vulnerable



Middle person attacks on NTLM

- NTLM doesn't protect against Middle Person attacks
- The most common tool for demonstrating middle person attacks on NTLM is SMBRelay
- SMBRelay is complicated, and works when people connect to shares
- I decided to exploit the willingness of browsers to transparently authenticate with local machines to connect back over SMB.
- Browser's are used very frequently, not a lot of waiting for share connecting

Middle person attacks on NTLM

- This means I am using NTLM over HTTP authentication to access an NTLM protected SMB resource (a share)
- People use web browsers as administrator frequently, and it's easy to tell when they are doing it – for example going to windows update is a give away.
- My code speaks HTTP and SMB – thanks to jCIFS!

NTLM middle person attack explained

- **Attacker may do DNS or Netbios response forging to acquire victims**
- **Victim connects to a local “attack web server” over HTTP**
- **Victim sees attack server as in the “Local Intranet Zone”**
 - Allows automatic NTLM authentication
- **Attack server connects to victim or target resource over SMB**
 - Victim must have file sharing accessible, be logged in as an account with access, support NTLM and not require SMB signing
- **Attack server gets a challenge from the victim**
- **Attack server demands NTLM authentication and reflects the challenge to the victim**
- **Victim responds correctly to the challenge**
- **Attack server provides the victim’s response and authenticates**
- **Attack server copies the backup SAM or does whatever else it likes**
- **Attack server serves a page to the victim or redirects them**

NTLM middle person attack explained

I have two versions

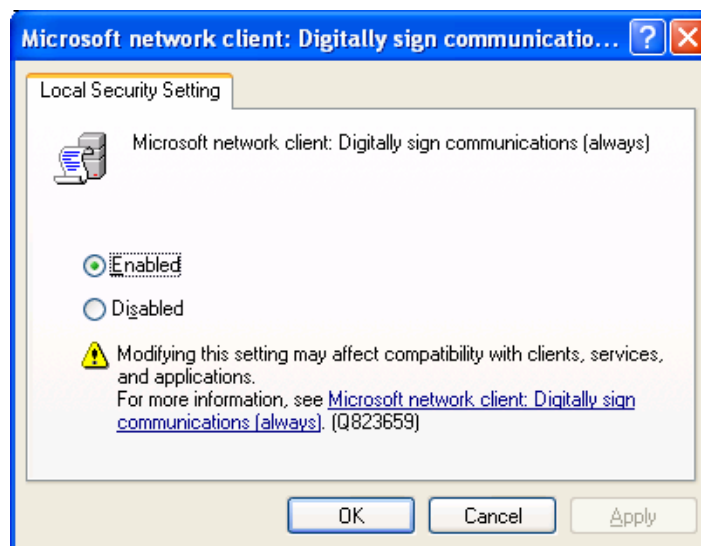
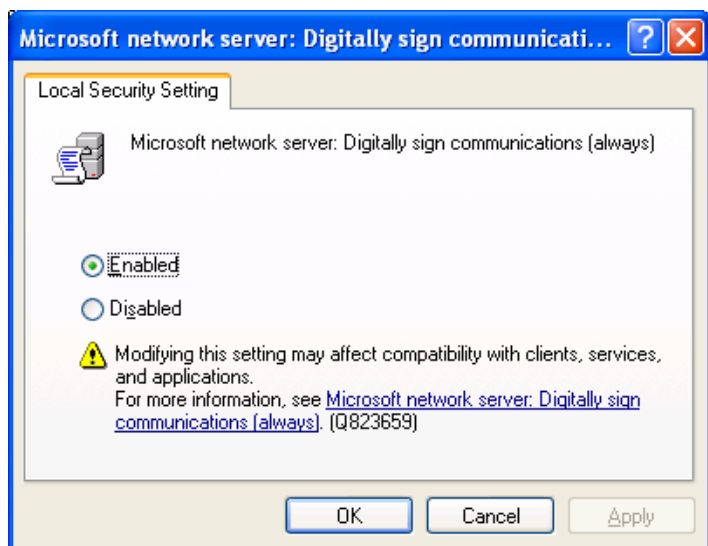
- **One grabs the backup SAM from admin\$/repair/sam**
 - It often contains the NTLM Hash for the administrator
 - It is often useable on other machines on the network
 - It can be used directly to authenticate with the tool I demoed earlier
- **The other connects and launches my “modified SmbShell”**

The second version is more fun to watch, and is more flexible in terms of exploit potential

DEMO

Protecting against NTLM middle person attacks

- Use NTLM authentication on web servers only with SSL
- Disable automatic NTLM authentication in browsers
- Setting must be to “always” use SMB Signing. Negotiating signing does nothing to protect you. Exploiting this vulnerability requires an active attacker, and active attackers will just downgrade if they have the option. Change the settings from default – this ships vulnerable



Take away's

I hope you agree that

- **NTLM is too dangerous to use on your network**
 - Use NTLMv2 or Kerberos, and set them to “always”, otherwise attackers just downgrade
 - Use SMB Signing to prevent middle person attacks
 - Only use NTLM or NTLMv2 auth on websites when using HTTPS
- **IE should be set not to automatically NTLM authenticate**
- **Never use identical administrator passwords across machines as they are stored in password equivalent form**
- **Samba shares are no less vulnerable than Windows**
- **Sorry, but I don't have a solution for Mac OS X - ADmitMac (<http://www.thursby.com/>) has one I haven't tried**
- **Future authentication protocols should use Zero-Knowledge Proof of Knowledge techniques like SRP does.**
- **iSEC Partners has cool, unique [tools](#) to clearly demonstrate the practical importance of common risks.**

Questions?

Shameless plug

[Information Security Partners](https://www.isecpartners.com) can help your organization understand and mitigate its application or network security risks. We perform protocol, application, and network penetration testing, training and complimentary security services.

See iSEC Partners online at

<https://www.isecpartners.com>

Or talk to the author: Jesse Burns, Principal Partner

Email: Jesse at iSECPartners dot com