Nail the Coffin Shut:

NTLM IS DEAD

Kurt Grutzmacher - Defcon 16
grutz @ jingojango.net

Who I am...

http://grutztopia.jingojango.net/

Corporate Penetration Tester for nearly a decade. Worked in the financial sector for a long time and now breaking the utility sector (SCADA!)

Dabbler in Metasploit development, getting Free MacWorld passes, half-price LinuxWorld passes, security research and spreading the good word of OWASP

Why this matters

As enterprises started to secure their internal networks we kept finding less and less when it came to "remotely exploitable" systems.

After shifting to more Web Security tests the question came up: "What can you do with an XSS inside the corporate network?"

Answer: How about owning the Domain? (yes, really)

Acronym Hell
&
Protocol Details

Quick Definitions

LM - *LAN Manager*

- Really old and really tired, never use this again
- Finally disabled by default in Vista and Server 2008

NTLM - *NT LAN Manager*

- Replaced "LAN Manager" (for a good reason)
- A "suite" of protocols for authentication and security: "NTLM Security Support Provider (NTLMSSP)"
- Also known as "ntlm 0.12"
- Describes an authentication protocol *and* the hash result

Kerberos - *Kerberos*

- But not *just* Kerberos, Microsoft's **extended** Kerberos!

Scrabble take two

Nonce - *Number Used Once*

Used to defeat replay attacks

SSPI - *Security Support Provider Interface*

Microsoft API to several security routines

SPNEGO - *Simple and Protected GSSAPI Negotiation Mechanism*

I don't know what to speak to you, so lets negotiate!

IWA - *Integrated Windows Authentication*

The act of negotiating authentication type using SPNEGO

Windows Authentication

NTLM Authentication Protocol is a challenge-response scheme that can be broken into three "Types":

- *Type 1*: Client sends "Hi, I want to talk to you"
- *Type 2*: Server sends "Ok, here are the various features and protocols I support including a *nonce* for you to encrypt your hashes with so nobody can replay it later in case they capture it. Oh and the domain you should authenticate to."
- *Type 3*: Client response "Sweet, I agree on the features you desire and support them in my daily life. Here's the username, domain again, workstation name, and the encrypted LM and NTLM hashes."

The server recovers the LM/NTLM hashes and compares them to its internal table and grants / denies accordingly in the response to a Type 3 message.

LM is a rotted corpse

1. Password converted to upper case
2. Password is null-padded or TRUNCATED to 14 bytes
3. Password is split into two halves of 7 bytes each
4. Two DES keys are created, one from each 7 byte half:
   4.1. Convert each half to a bit stream
   4.2. Insert a zero bit after every 7 bits
5. Each key DES-encrypts the string “KGS!@#$%” creating two 8 byte ciphertext values
6. Concatenate the two results for your LM hash

NTLMv1 Protocol is...

1. Cleartext is converted to Unicode and hashed with MD4 -- This is the "NTLM Hash"
2. The 16-byte hash is null-padded to 21 bytes and split into three 7-byte values
3. These values are each used to create three DES keys
4. Each of these keys is used to DES-encrypt the nonce from the Type 2 message, resulting in three 8-byte ciphertext values
5. These three ciphertext values are concatenated to form a 24-byte value which goes into the Type 3 response.

NTLMv2 Protocol is...

1. NTLM hash is generated (MD4(unicode(cleartext)))
2. Unicode uppercase username and domain name are concatenated
3. An HMAC-MD5 of the NTLM hash and result from Step 2 is made
4. A blob is created using the timestamp, a client nonce and static data
5. An HMAC-MD5 of the blob and result from Step 3 is made
6. This 16-byte value result is now the NTLM hash for use in the authentication protocol.

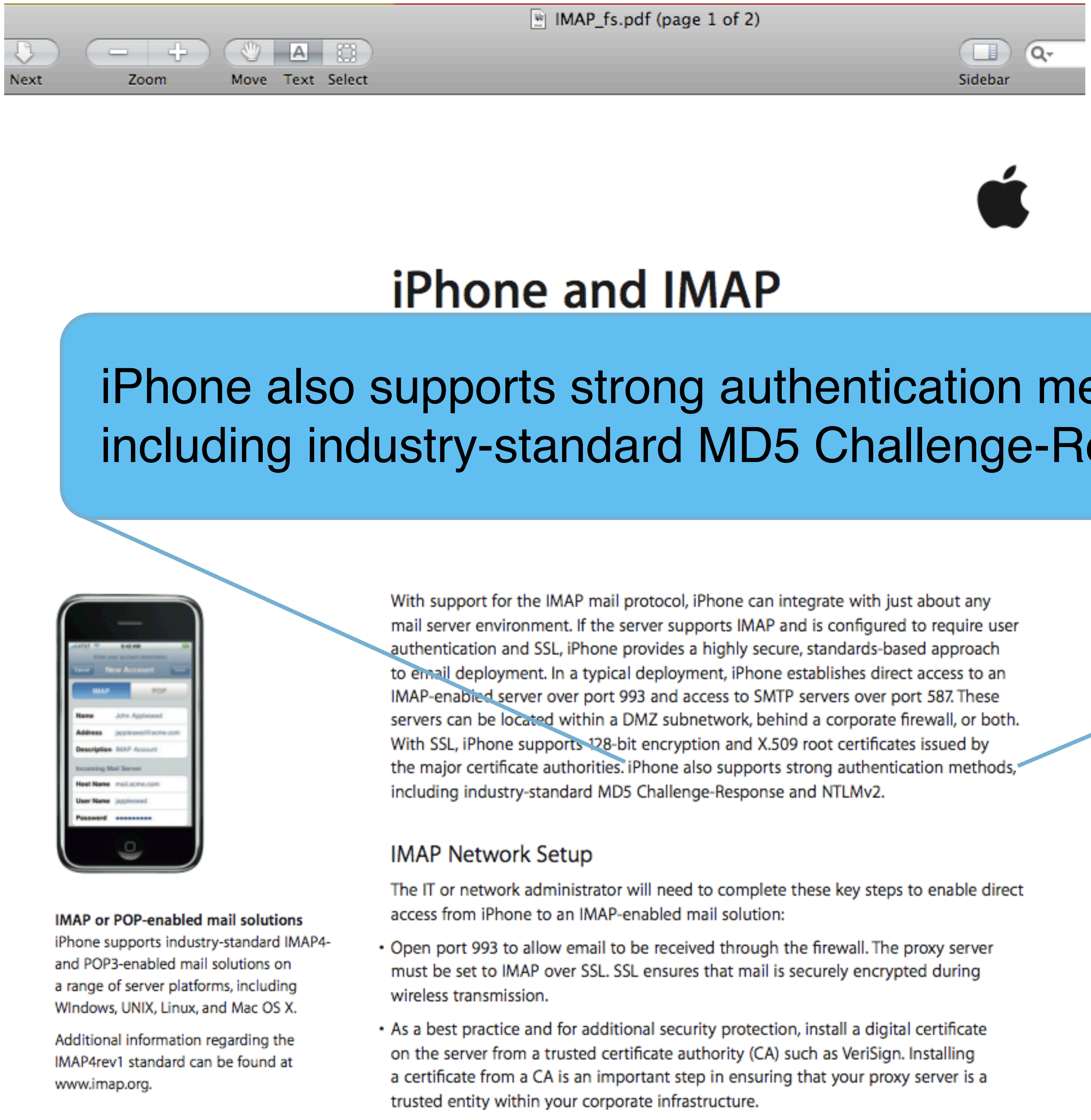
NTLMv2 Session...

1. An 8-byte client nonce is generated and padded to 24 bytes
2. The result is placed into the LM field of the Type 3 response -- No LM result is generated or passed using NTLMv2 Session
3. Server's nonce is concatenated with the client nonce -> Session nonce
4. Session nonce is MD5'd and truncated to 8 bytes -> Session hash
5. NTLM hash is generated, null padded to 21 bytes and split into three 7-byte values
6. These values are each used to create three DES keys
7. Each of these keys is used to DES-encrypt the nonce from the Type 2 message, resulting in three 8-byte ciphertext values
8. These three ciphertext values are concatenated to form a 24-byte value which goes into the Type 3 response.

and NTLM is supported...

...everywhere!

- in Microsoft products (IIS, IAS, Exchange, Internet Explorer)
- in Samba and Apache and PAM
- in other browsers (Mozilla Firefox and Safari)
- in proxy servers to support browsers who don’t do NTLM
- in your iPhone (really!) for Enterprises
- in OSX to connect to Windows shares
- in WinCE to connect to Windows shares
- in ToasterBrandConsumerDevice to connect to Windows shares
- in * to connect to Windows shares



iPhone and IMAP

With support for the IMAP mail protocol, iPhone can integrate with just about any mail server environment. If the server supports IMAP and is configured to require user authentication and SSL, iPhone provides a highly secure, standards-based approach to email deployment. In a typical deployment, iPhone establishes direct access to an IMAP-enabled server over port 993 and access to SMTP servers over port 587. These servers can be located within a DMZ subnetwork, behind a corporate firewall, or both. With SSL, iPhone supports 128-bit encryption and X.509 root certificates issued by the major certificate authorities. iPhone also supports strong authentication methods, including industry-standard MD5 Challenge-Response and NTLMv2.

IMAP Network Setup

The IT or network administrator will need to complete these key steps to enable direct access from iPhone to an IMAP-enabled mail solution:

- Open port 993 to allow email to be received through the firewall. The proxy server must be set to IMAP over SSL. SSL ensures that mail is securely encrypted during wireless transmission.
- As a best practice and for additional security protection, install a digital certificate on the server from a trusted certificate authority (CA) such as VeriSign. Installing a certificate from a CA is an important step in ensuring that your proxy server is a trusted entity within your corporate infrastructure.

IMAP or POP-enabled mail solutions

iPhone supports industry-standard IMAP4- and POP3-enabled mail solutions on a range of server platforms, including WIndows, UNIX, Linux, and Mac OS X.

Additional information regarding the IMAP4rev1 standard can be found at www.imap.org.

iPhone also supports strong authentication methods, including industry-standard MD5 Challenge-Response and ***NTLMv2***.

Seems strong...

NTLM is better than LM (well, duh):

1. Cleartext is NOT converted to upper case
2. Passwords are NOT broken into blocks of 7 bytes
3. DES not so good but it's the last step to generate results and client/server nonces protect from pre-computed attacks
4. Server nonces do ***not*** protect pre-computed attacks however.

In the grand scheme of things the LM and NTLM hashes should be considered the same as cleartext passwords. When obtained an attacker does not need to find the cleartext in order for them to be used.

So why is it dead?

NTLM has shown its survivability by hanging on to “backwards compatibility” and ubiquitous deployment. If it’s everywhere what is the incentive to get rid of it?

You’ve got:

Replay protection

Mixed case support from cleartext to ciphertext

Server nonces in v1

Client and Server nonces in v2

Message digests

Timestamps

..sounds good so far!

Lets not get ahead of ourselves just yet.

In an ENTERPRISE we have the joyful tune of “Single Sign-On”. When a workstation becomes a member of the domain any user that logs on can access their resources with only having to type their password once during the log on process

This means that the cleartext or LM/NTLM ciphertext may be stored within the memory of the workstation throughout the session or beyond!

It also means that authentication can happen at the request of an application and not by a user.

2

Past Attacks (in a nutshell)

Our Threat Model

The NTLM protocols pre-suppose an Enterprise authentication system using Windows Domains or Active Directory.

Evildoers must fit within this environment in order to take advantage of it so they usually have to have inside access.

Doesn’t mean this isn’t an external threat, just that at this time I can’t think of or have seen an attack from the outside in.

Protocol Downgrade

During SPNEGO the client gets the first word on protocol support:

Signing, Sealing, use NTLM, Always Sign, send Target block, etc.

The server responds with their own list of support:

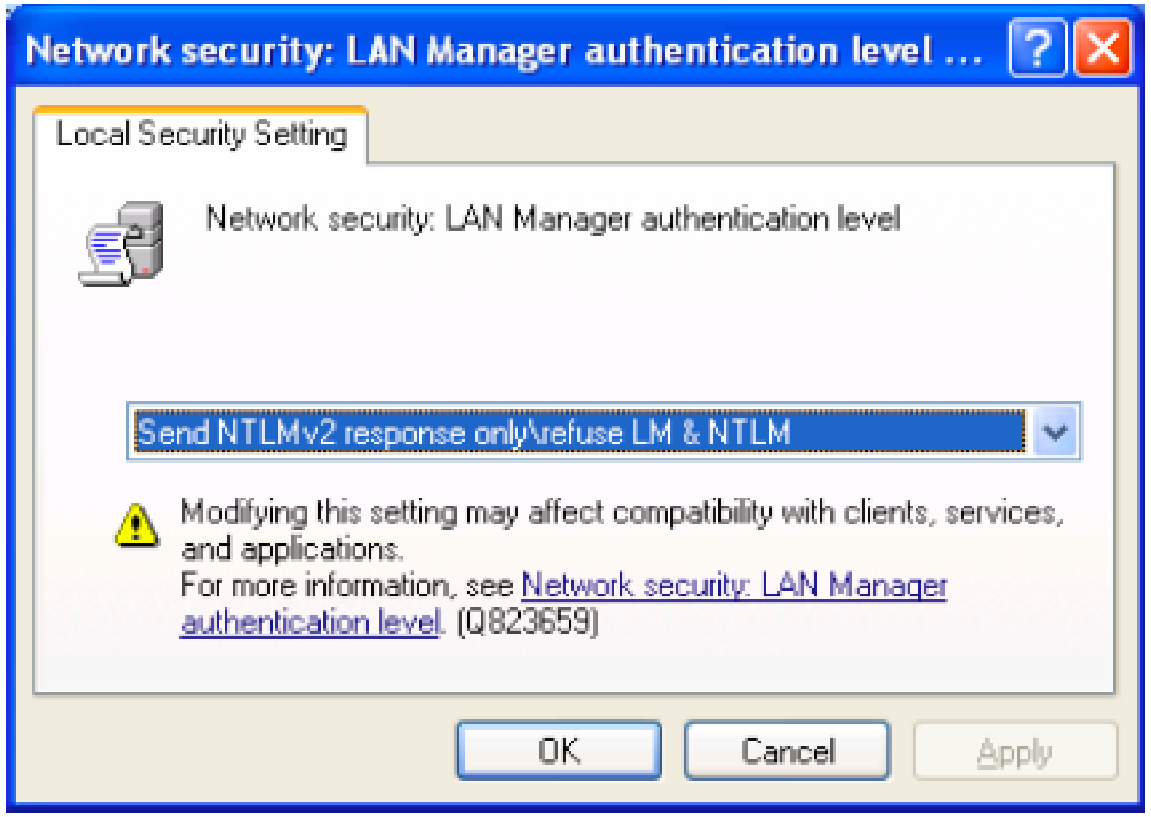
NTLM2 key, Target block included, 128-bit encryption, etc

If both sides agree the client sends all the requisite data for an authentication attempt and waits for a response.

Using MITM tools such as Cain & Able or Ettercap an attacker can force either side to negotiate LOWER than they would have otherwise.

Protecting Downgrade

Through a GPO or within Local Security Policy change your LAN Manager authentication level:



Network security: LAN Manager authentication level ...

Local Security Setting

Network security: LAN Manager authentication level

Send NTLMv2 response only\refuse LM & NTLM

Modifying this setting may affect compatibility with clients, services, and applications.
For more information, see Network security: LAN Manager authentication level. (Q823659)

Replay Attacks

Comes in two forms:

- Network capture and replay if no nonce
- Obtaining the LM/NTLM hashes and using them during auth

“Pass The Hash” is the term and it is pure awesome:

- Obtain privileges on a server or workstation
- Dump a copy of stored hashes (SAM, LSASS, running processes)
- Skip the part of “converting to LM/NTLM” during the Network authentication routines
- Who needs to crack hashes anymore?

Tools for Replay

Obtain hashes:

- FGDump -
- PWDumpX -
- Cain & Able -
- Pass The Hash Toolkit
- Metasploit, Canvas, CORE Impact

Passing The Hash

- Hydra
- Patches for Samba from JoMoKun (sorry fizzgig)
- Pass The Hash Toolkit
- Metasploit, Canvas, CORE Impact

SMB Relay (original)

http://www.xfocus.net/articles/200305/smbrelay.html

First released in March, 2001 at @tlantaCon by Sir Dystic of cDc

Listens for NBT requests and collects LM/NTLM hashes for cracking

Version 1:

- Connects back to the requester using their credentials
- Emulates an SMB server for the attacker to connect to
- TCP/IP Addresses only
- Generally great for one-off attacks

Version 2:

- Supported NetBIOS names
- Relay to a third-party host

SMB Relay (Metasploit)

Re-engineering of SMB Relay script as a Metasploit attack module

Metasploit already had LM/NTLM hash capture support since 2.7

Can connect back to original host or forward to a single host

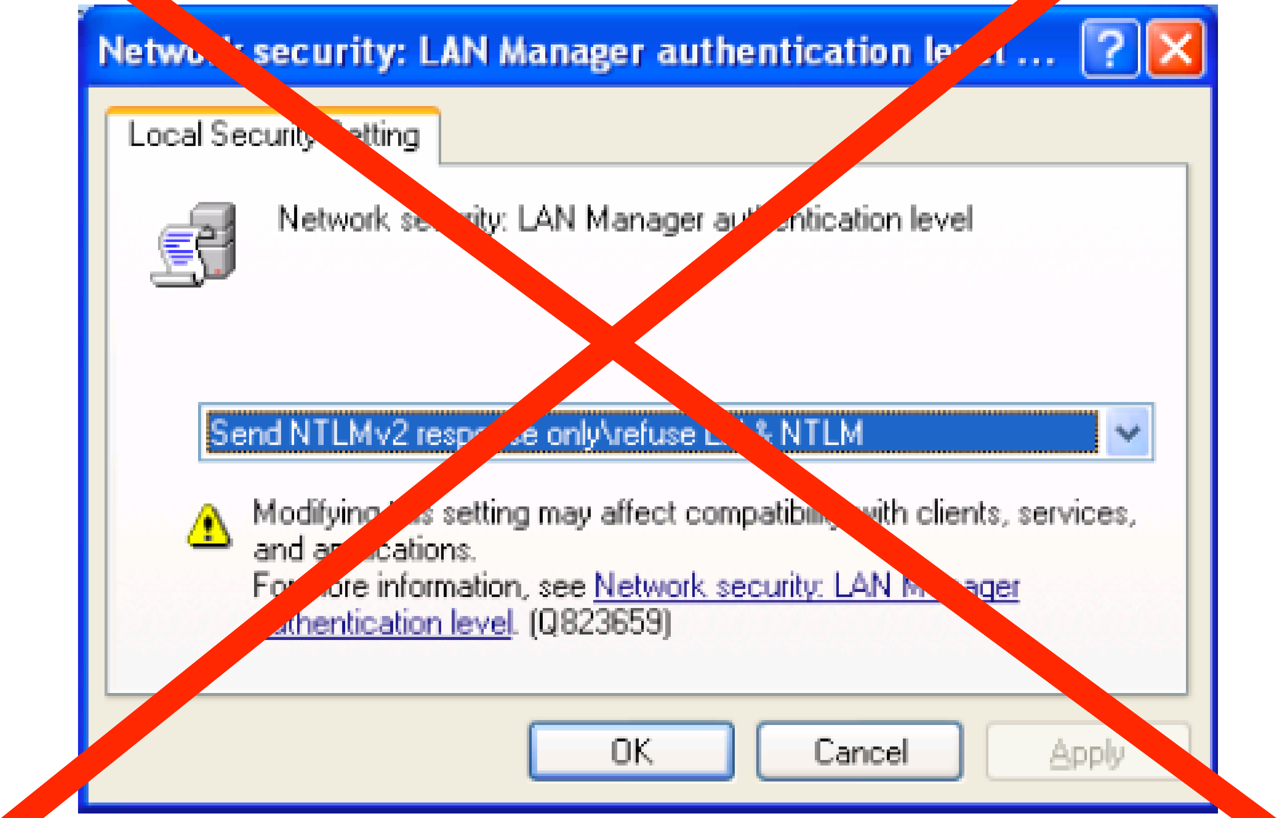
Works **great** if:

- Users are local administrators
- Server service has been started on their workstations
- or the users have rights to your destination host

See last year's "Tactical Exploitation" presentation for other cool ideas.

Stopping SMBRelay

Through a GPO or within Local Security Policy change your LAN Manager authentication level:



Network security: LAN Manager authentication level ...

Local Security Setting

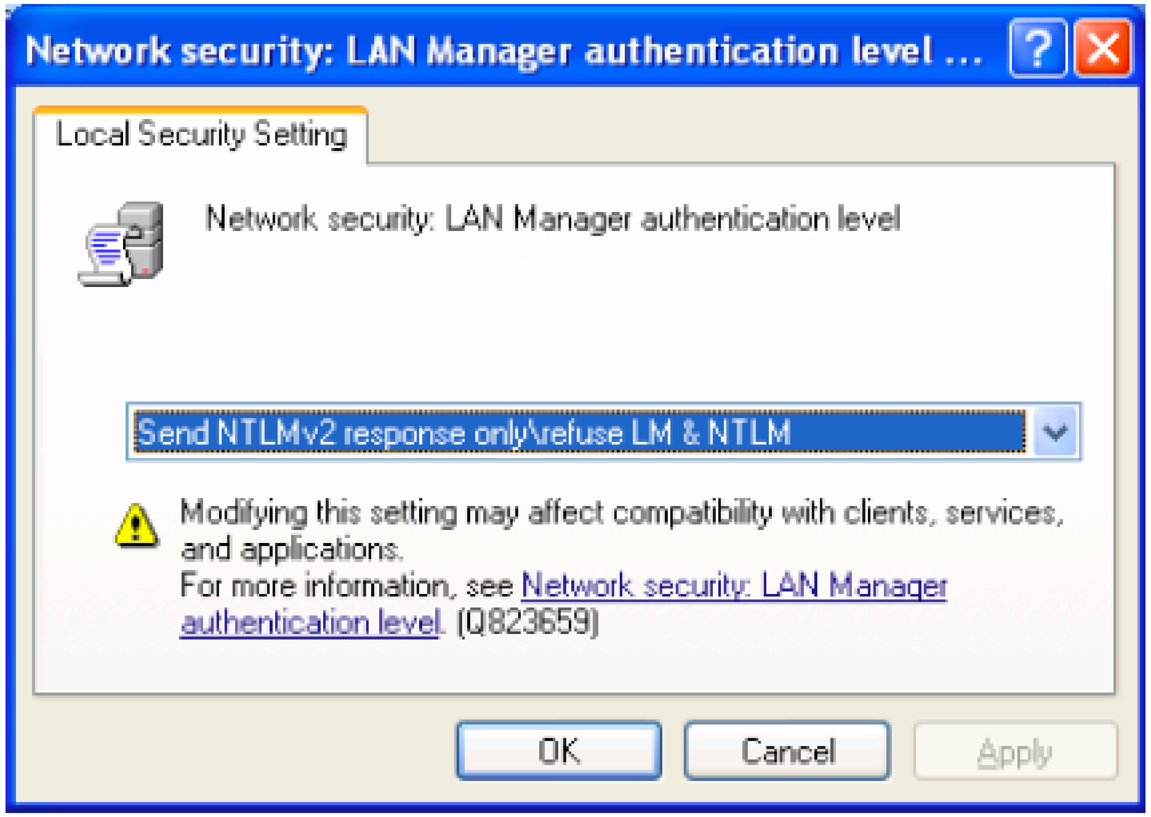
Network security: LAN Manager authentication level

Send NTLMv2 response only\refuse LM & NTLM

Modifying this setting may affect compatibility with clients, services, and applications.
For more information, see Network security: LAN Manager authentication level. (Q823659)

...but not really

NTLMv2 will not stop this attack, it's just that MSF doesn't fully support the protocol yet.



Network security: LAN Manager authentication level ...

Local Security Setting

Network security: LAN Manager authentication level

Send NTLMv2 response only\refuse LM & NTLM

Modifying this setting may affect compatibility with clients, services, and applications.
For more information, see Network security: LAN Manager authentication level. (Q823659)

3

New variant to old attacks

NTLM over ...

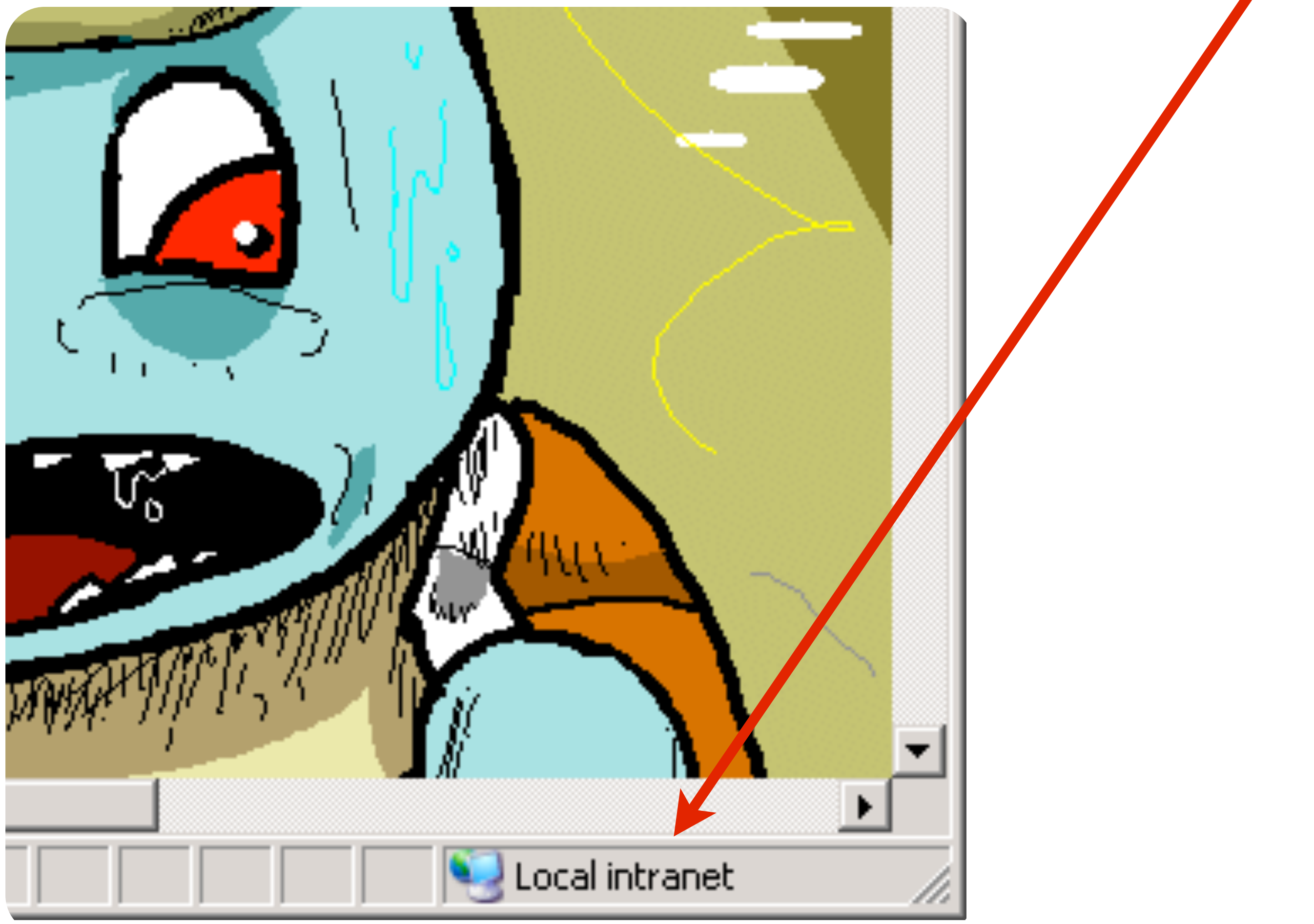
HTTP, IMAP, POP3, SMTP, NNTP, etc. . .

While NTLM is a Microsoft protocol, in order to fully support SSO it has to support standard protocols. NTLM "Type Messages" are Base64 encodings of the NTLM protocol to transmit over 7-bit protocols.

Part of the Integrated Windows Authentication suite.

IE Trust Zones

In order for Internet Explorer to perform Integrated Windows Authentication the browser must be in the "Local Intranet" or a customized zone.



Automatic Authentication

http://support.microsoft.com/kb/258063

The following conditions must be met for Internet Explorer to automatically authenticate a user's logon and password and maintain security:

- Windows Integrated authentication, also known as Windows NT Challenge/Response, must be enabled in the Web site properties in IIS. Anonymous authentication is attempted first, followed by Windows Integrated authentication, Digest authentication (if applicable), and finally Basic (clear text) authentication.
- Both the client and the Web server must be either in the same Microsoft Windows NT-based or Microsoft Windows 2000-based domain or in trusted Windows NT-based or Windows 2000-based domains in which the user's account can be granted permissions to resources on the IIS-based computer.
- The user's browser must be Internet Explorer. Internet Explorer is the only browser that supports Windows Integrated authentication (NTCR).
- Internet Explorer must consider the requested URL to be on the intranet (local). If the computer name portion of the requested URL contains periods (such as http://www.microsoft.com and http://10.0.0.1), Internet Explorer assumes that the requested address exists on the Internet and does not pass any credentials automatically. Addresses without periods (such as http://webserver) are considered to be on the intranet (local); Internet Explorer passes credentials automatically. The only exception is addresses included in the Intranet zone in Internet Explorer.

Forcing Trust Zones

It has been possible in the past to force IE into the Local Intranet zone through the use of Flash or Java applets.

http://heasman.blogspot.com/2008/06/stealing-password-hashes-with-java-and.html

Mozilla Auth Setup

Firefox supports NTLM! In about:config

Enable IWA:

```
network.negotiate-auth.trusted-uris: list,of,uris
network.negotiate-auth.using-native-gsslib: true
```

Or just NTLM:

```
network.automatic-ntlm-auth.trusted-uris: list,of,uris
network.ntlm.send-lm-response: false
```

Firefox 3.0 does honor IE's Trust Zones but not for IWA.

NTLM in <browser>

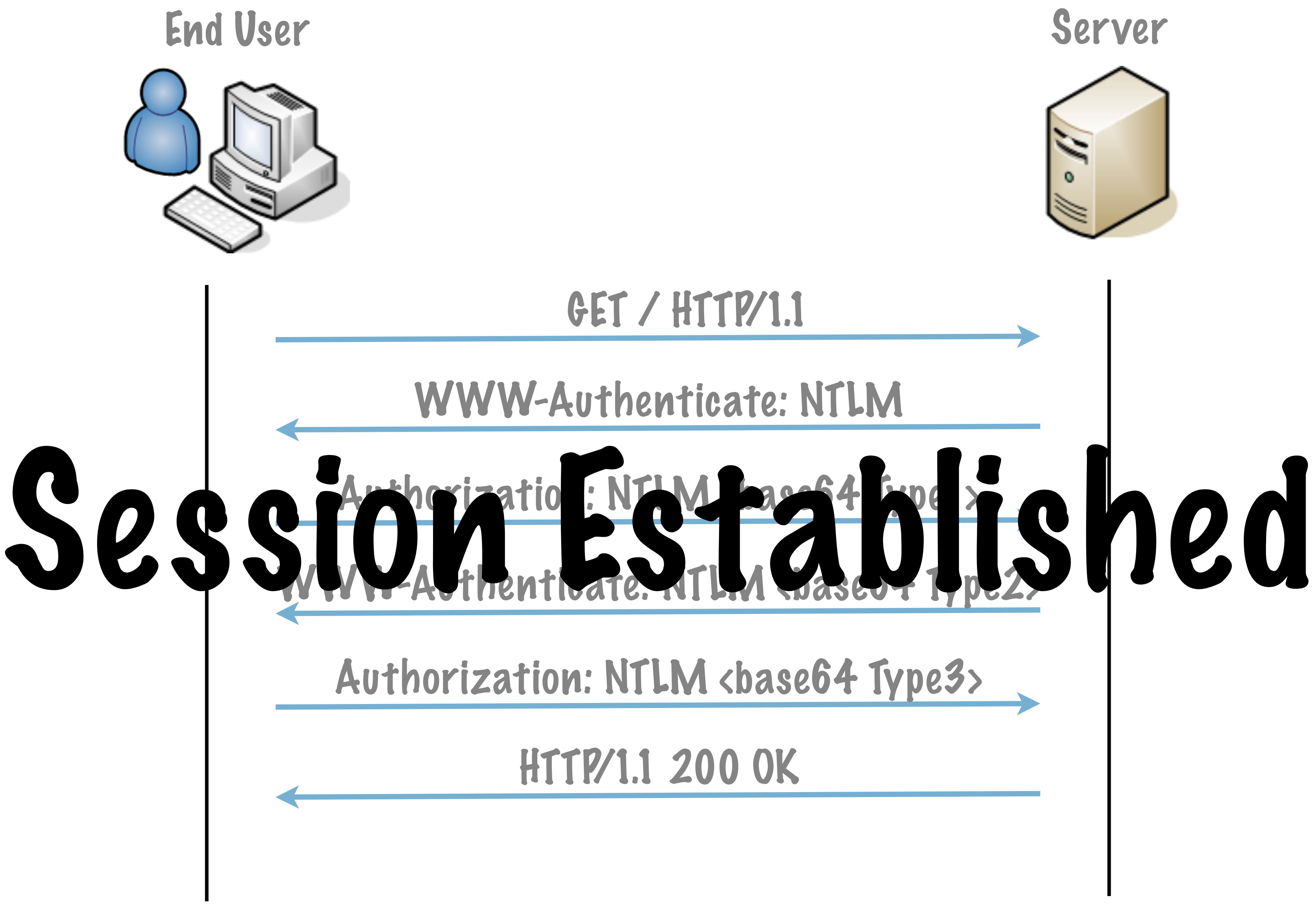
Opera does not support NTLM authentication directly, you must go through a proxy server.

Safari for Windows will do NTLM but does not do Integrated Windows Auth. Typically a proxy server is used for OS X or stored credentials in the keychain.

Wget/CURL both support NTLM on the command line.

Links/Lynx … why? maybe, not something I checked - usually use a proxy server like NTLMAPS.

HTTP NTLM Auth



End User

Server

GET / HTTP/1.1

WWW-Authenticate: NTLM

Authorization: NTLM <base64 Type1>

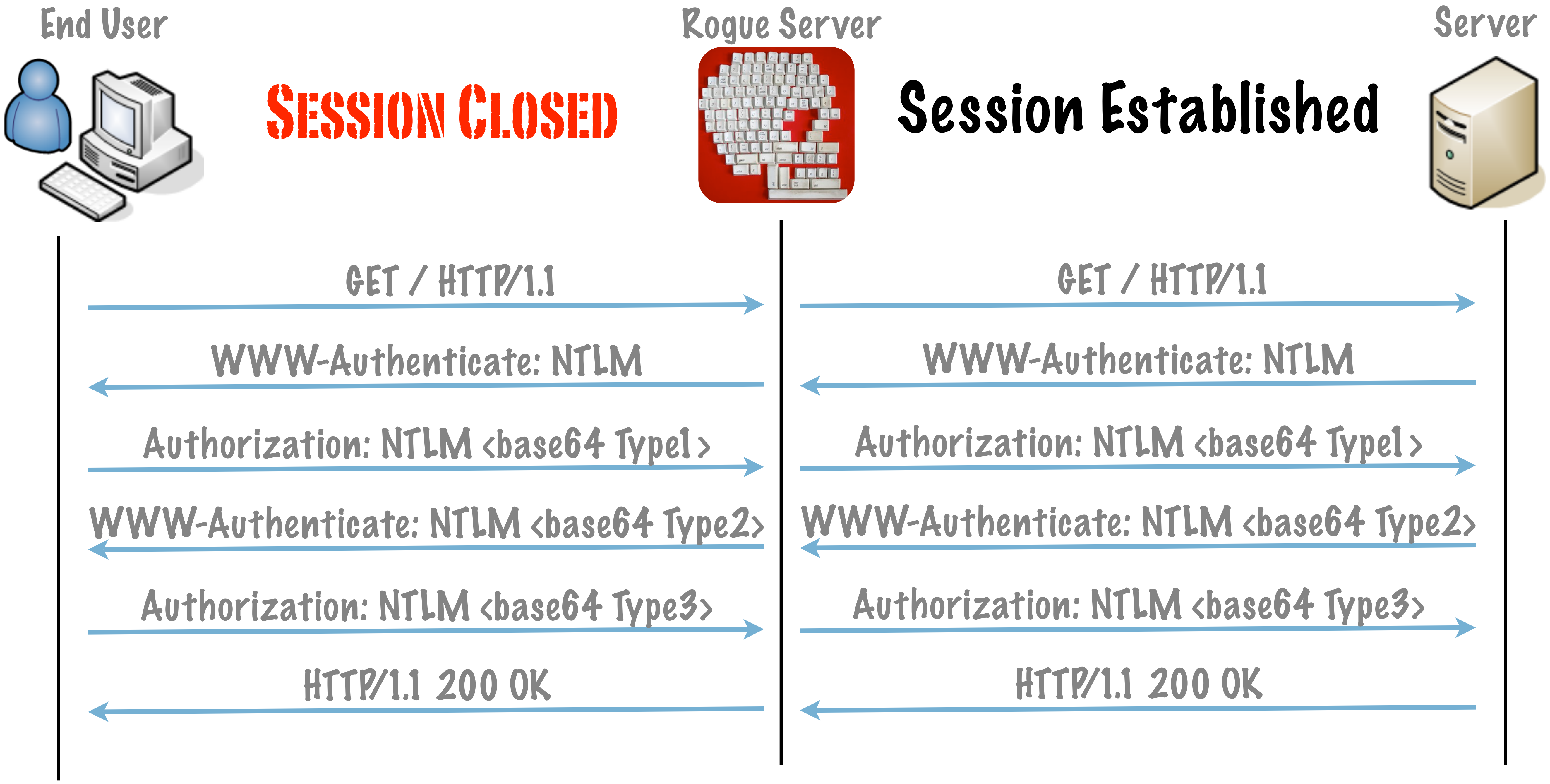
WWW-Authenticate: NTLM <base64 Type2>

Authorization: NTLM <base64 Type3>

HTTP/1.1 200 OK

Session Established

Monkey In The Middle?



End User

Rogue Server

Server

SESSION CLOSED

Session Established

GET / HTTP/1.1

WWW-Authenticate: NTLM

Authorization: NTLM <base64 Type1>

WWW-Authenticate: NTLM <base64 Type2>

Authorization: NTLM <base64 Type3>

HTTP/1.1 200 OK

GET / HTTP/1.1

WWW-Authenticate: NTLM

Authorization: NTLM <base64 Type1>

WWW-Authenticate: NTLM <base64 Type2>

Authorization: NTLM <base64 Type3>

HTTP/1.1 200 OK

What does this mean?

As a **Rogue Server** we're able to bridge authentication requests using Type Messages by directing HTTP requests to us (<img src>)

NTLMv2 messages pass through without modification

This was previously possible via SMBRelay but very limited target scope using WPAD+SOCKS or forcing file:// or smb:// connections

Jesse Burns @ iSEC Partners described the HTTP->SMB link in 2004 but never released source code

In late 2007 I implemented a hash collector and HTTP->POP3 bridge

Earlier this year Eric Rachner released "scurvy"

Introducing Squirtle

Squirtle? What the...

Squirtle is a **Rogue Server** with **Controlling Desires!** It does not require:

- Man In The Middle techniques such as:
  - ARP Poisoning
  - DNS Redirection
  - GRE Tunnels

Squirtle does require:

- The browser be in a "trusted zone" for IWA to work
- Support for WWW-Authenticate: NTLM
- You somehow direct the browsers to it (XSS, proxy, <img>, etc)

Implements "Pass The Dutchie" attacks:

- Dutchie == Server requirements and client response

Passing The Dutchie

Clients are given a pre-computed session key during their first connection to Squirtle in a browser cookie. All requests reference this key.

keepalives are sent from the Client to the Squirtle controller for action requests. The timeout is specified within the config file.

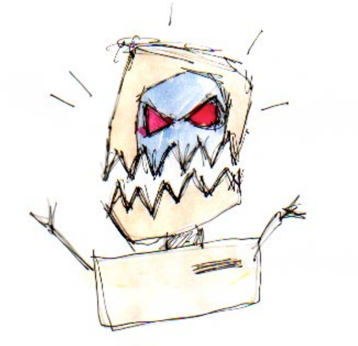
Evil Agents can request specific clients authenticate with a given nonce or specific variables.

Evil Agent requests are “protected” via basic-auth.

Data and Session State is logged and maintained in a SQL database (sqlite, mysql or postgres)

Passing The Dutchie

Evil Agent



Squirtle

Controlled Clients

User A
Session 1
User A User B

Session 1 Auth with this Nonce
Type 1 3 Response [illegible]

NTLM Required [illegible]ating

SQUIRTLE!
SQUIRTLE!
SQUIRTLE!

Auth[illegible] Nonce

Negotiation [illegible]
Send [illegible] Nonce
Completed
Awaiting Response!

XSS on Homepage [illegible] Squirtle

Session	Username	Req Type	Type2 Response
1	A	Type2 Nonce	Type2 Message Hashes
2	B	Nonce	Hashes
3	C	Nonce	Hashes

Enterprise Server

Corporate Homepage

What does this mean?

Past attacks against Windows authentication have been either directed at a single server or back towards the client.

By corralling clients and exposing an API to externally written tools, Squirtle allows proxy servers to be written in any language that the attacker desires. They don’t need to worry about grabbling clients and holding on to them, let Squirtle do that.

Existing frameworks such as Metasploit, Canvas and CORE Impact can use Squirtle to perform attacks against resources that require authentication without having to obtain cleartext or LM/NTLM hashes!

Thinking about it...

I came up with this particular scenario and tool because after finding a ton of internal XSS vulnerabilities the response was “Great, you can run a port scanner or send print jobs. What else?”

So think about this:

Internal servers with web programming errors (XSS, SQL, etc)

Open SMB c:\inetpub\www shares with write access

An internal PHPNuke or Blog or Other “Open Source” Scripts

Sending an E-mail with a link inside of it

The evil act of opening Microsoft Office documents

They will all be controlled by the mighty Squirtle!

Evil Agent Functions

URI	***Description***
/controller/listsessions/	List all session states
/controller/allhashes/	List all hashes
/controller/allusers/	List all users
/controller/listuser/	List a specific user
/controller/session/	List a specific session
/controller/redirect/	Force a redirect (lose the client)
/controller/static/	Request static auth
/controller/type2/	Request response to Type 2 msg
/controller/clearsession/	Clear a session state

Demonstrations

Squirtle -> Web Proxy

1. Modified “ntlmaps” proxy to talk to Squirtle when web server requests NTLM authentication.
2. Sends Squirtle the Type 2 message for processing using an attacker-specified user.
3. Squirtle builds the request, waits for the keepalive timeout, then requests the controlled browser authenticate.
4. Resulting Type 3 message is delivered back to Squirtle and then passed along to the server to complete authentication.
5. NTLMv2 APPROVED!

DEMO!

Squirtle -> smbclient

1. Modified “smbclient.py” tool from IMpacket to talk to Squirtle for NTLM authentication.
2. Sends Squirtle the NTLM flags and domain information or processing with an attacker-specified user.
3. Squirtle builds the request, waits for the keepalive timeout, then requests the controlled browser authenticate.
4. Resulting Type 3 message is delivered back to smbclient where it’s unpacked and placed into SMB fields.
5. SMB packet delivered to server to complete authentication.

DEMO! (not yet... sigh)

More to do. . .

Proxy integration:

- NTLMAPS (99% complete, needs testing, management interface)
- SMBClient (from IMPACKET -> 80% complete)
- up-imapproxy (70% complete)
- Proxies that keep TCP connections open are awesome!

Administrative web or Flex/Air interface

Additional integration with MySQL, Postgress, etc

Real-world testing!

http://code.google.com/p/squirtle

5 Defenses

Stop the pain!

For Web Servers:

- Turn off support for "WWW-Authenticate: NTLM"
- Enforce Kerberos authentication

For SMTP, IMAP, NNTP, everything else etc.

- Enable Kerberos authentication

For all:

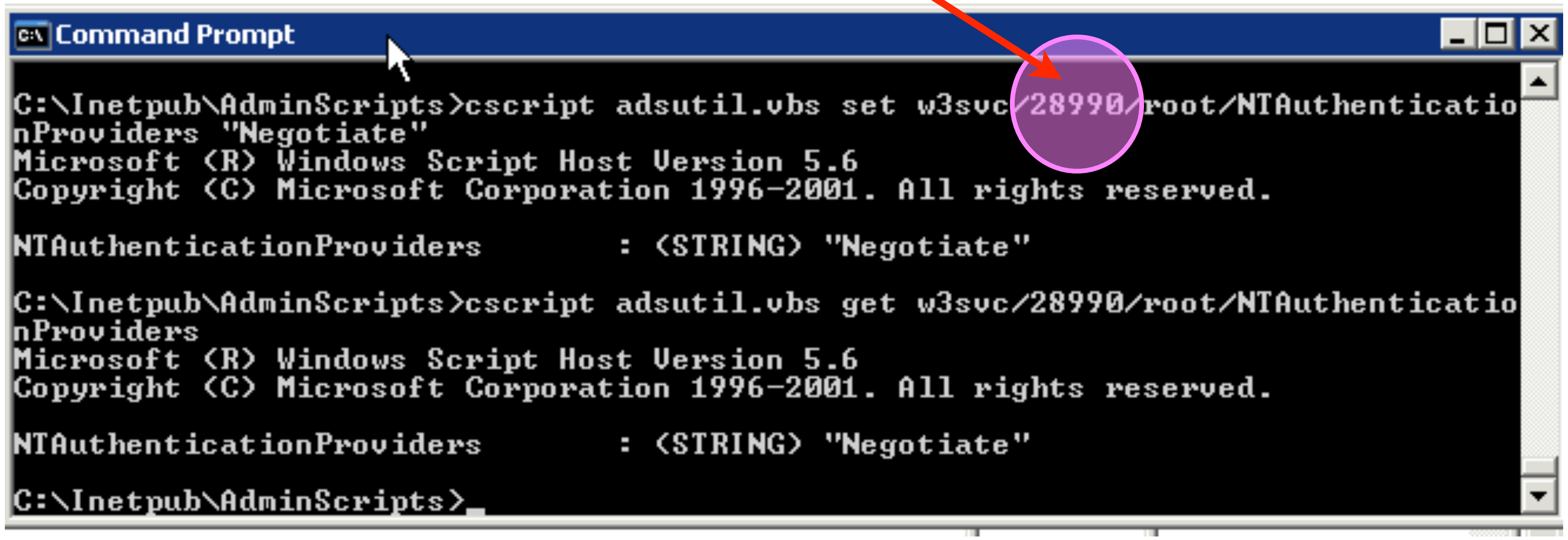
- Require message signing for all communication at the very least.

Will Kerberos hold? Only time will tell but it's been pretty good so far.

Protecting IIS

For each instance, follow http://support.microsoft.com/kb/215383

cscript adsutil.vbs set w3svc/instance#/root/NTAuthenticationProviders "Negotiate"



```
C:\Inetpub\AdminScripts>cscript adsutil.vbs set w3svc/28990/root/NTAuthenticatio
nProviders "Negotiate"
Microsoft (R) Windows Script Host Version 5.6
Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.

NTAuthenticationProviders        : (STRING) "Negotiate"

C:\Inetpub\AdminScripts>cscript adsutil.vbs get w3svc/28990/root/NTAuthenticatio
nProviders
Microsoft (R) Windows Script Host Version 5.6
Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.

NTAuthenticationProviders        : (STRING) "Negotiate"

C:\Inetpub\AdminScripts>_
```

This will break NTLM-only supported systems like NTLM proxies so do your testing before hand.

Forcing the client

Not possible. If a browser is in the Local Intranet zone and sees the "WWW-Authenticate: NTLM" header they will attempt to authorize with it.

Best bet at the moment is to enable NTLMv2-only and get rid of all your Windows NT servers (you have gotten rid of them, right?   RIGHT?)

At least with NTLMv2 decryption will take a long time should an attacker obtain user authentication packets.

(definition of "long" may change over time)

Q&A - URLs

Squirtle: http://code.google.com/p/squirtle

Pass The Hash Toolkit: http://oss.coresecurity.com/projects/pshtoolkit.htm

FGDump: http://www.foofus.net/fizzgig/fgdump/

Cain & Abel: http://www.oxid.it/cain.html

Special thanks to natron and parity, two guys who kept things interesting!

http://grutztopia.jingojango.net/ -- http://www.metasploit.com/