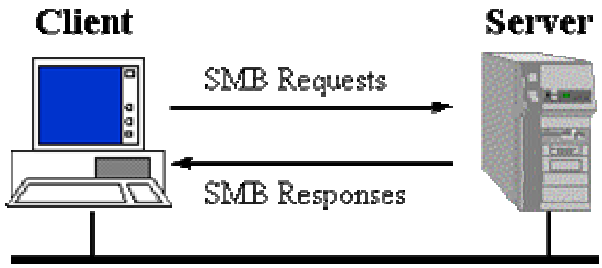
Security of SMB Protocol

SMB, **Server Message Block**, является протоколом **IBM** для предоставления общего доступа у файлам, папкам, принтерам, последовательным портам и коммуникационным абстракциям, таким как, именованные каналы и почтовые сегменты, между компьютерами. Этот протокол очень важен, так как сегодня большая часть персональных компьютеров уже имеют встроенную клиентскую и серверную реализацию этого протокола. Все ныне существующие версии Windows поддерживают клиентскую, а некоторые версии и серверную, реализацию SMB протокола. Первым документом, описывающим протокол, является документ **"IBM PC Network SMB Protocol"** от 1985 года компании IBM. Следующий документ был опубликован компанией **Microsoft** в 1987 году и назывался **"Microsoft Networks/OpenNET-FILE SHARING PROTOCOL"**. В дальнейшем протокол последовательно разрабатывался Microsoft и другими компаниями.

SMB – это клиент-сервер, запрос-ответ протокол, который предоставляет клиентским приложениям простой способ для чтения и записи файлов, а также запроса служб у серверных программ в различных типах сетевого окружения. Диаграмма справа показывает, как работает SMB. Единственное отличие от модели запрос-ответ (то есть модели, в которой, когда клиент посылает запрос и сервер отвечает ему ответом) это, когда клиент посылает в качестве запроса оппортунистические блокировки, а сервер вынужден отпустить уже предоставленную блокировку, так как другой клиент запросил открытие файла в режиме, несовместимом с предоставленной блокировкой. В этом случае, сервер посылает клиенту уведомительное сообщение о том, что блокировка была снята. Серверы предоставляют файловые системы и другие ресурсы (принтеры, почтовые сегменты, именованные каналы и т.д.) для общего доступа в сети. Клиентские компьютеры могут иметь у себя свои носители информации, но они так же хотят иметь доступ к ресурсам, предоставленным сервером для общего пользования.



Клиенты соединяются с сервером, используя протоколы **TCP/IP** (а точнее **NetBIOS** через TCP/IP), **NetBEUI** или **IPX/SPX**. После того, как соединение установлено, клиенты могут посылать команды серверу (эти команды называются SMB-команды или SMBs), который дает им доступ к ресурсам, позволяет открывать, читать файлы, писать в файлы и, вообще, выполнять весь перечень действий, которые можно выполнять с файловой системой. Однако, в случае SMB, эти действия совершаются через сеть. Как было сказано выше, SMB работает, используя различные протоколы. Следующая диаграмма показывает это:

<table>
<tr><td>OSI</td><td colspan="4"></td><td>TCP/IP</td></tr>
<tr><td>Application
Presentation</td><td colspan="4">SMB</td><td rowspan="2">Application</td></tr>
<tr><td>Session</td><td>NetBIOS</td><td rowspan="3">NetBEUI</td><td>NetBIOS</td><td>NetBIOS</td></tr>
<tr><td>Transport</td><td rowspan="2">IPX[1]</td><td rowspan="2">DECnet</td><td>TCP&UDP</td><td>TCP/UDP</td></tr>
<tr><td>Network</td><td>IP</td><td>IP</td></tr>
<tr><td rowspan="2">Link</td><td rowspan="2">802.2,
802.3,802.5</td><td>802.2</td><td rowspan="2">Ethernet V2</td><td rowspan="2">Ethernet V2</td><td rowspan="2">Ethernet or others</td></tr>
<tr><td>802.3,802.5</td></tr>
<tr><td>Physical</td><td colspan="4"></td></tr>
</table>

В сетевой модели OSI, протокол SMB используется как протокол Application/Presentation уровня и зависит от низкоуровневых транспортных протоколов. SMB может использоваться через TCP/IP, NetBEUI и IPX/SPX. Если TCP/IP или NetBEUI будут заняты, то будет использоваться NetBIOS API. SMB также может посылаться через DECnet протокол. Digital (ныне Compaq) сделала это специально для своего продукта PATHWORKS. NetBIOS, в случае использования через TCP/IP, имеет различные названия. Microsoft называет его в некоторых случаях NBT, а в некоторых NetBT. Так же встречается название RFCNB.

С начала существования SMB, было разработано множество различных вариантов протокола для обработки всевозрастающей сложности компьютерной среды, в которой он использовался. Договорились, что реальный вариант протокола, который будет использоваться клиентом и сервером, будет определяться командой **negprot (negotiate protocol)**. Этот SMB обязан посылаться первым до установления соединения. Первым вариантом протокола был **Core Protocol**, известный как SMB имплементация **PC NETWORK PROGRAM 1.0**. Он должным образом поддерживает весь набор основных операций, который включает в себя:

- коннект и дисконнект к файловым и принтерным ресурсам
- открытие и закрытие файлов
- открытие и закрытие принтерных файлов
- чтение и запись файлов
- создание и удаление файлов и директорий
- поиск директорий
- получение и установление атрибутов файла
- блокировка и разблокировка файлов

Следующие протоколы расширяют базовые возможности протокола:

SMB Protocol Variant	Protocol Name
PC NETWORK PROGRAM 1.0	Core Protocol
MICROSOFT NETWORKS 1.03	Core Plus Protocol
MICROSOFT NETWORKS 3.0	DOS LAN Manager 1.0
LANMAN1.0	LAN Manager 1.0
DOS LM1.2X002	LAN Manager 2.0
LM1.2X002	LAN Manager 2.0
DOS LANMAN2.1	LAN Manager 2.1
LANMAN2.1	LAN Manager 2.1
Windows for Workgroups 3.1a	LAN Manager 2.1?
NT LM 0.12	NT LAN Manager 1.0?
Samba	NT LAN Manager 1.0?
CIFS 1.0	NT LAN Manager 1.0

Некоторые вариации протокола ввели новые команды, некоторые просто изменили формат существующих команд и формат реакций сервера на команды (ответы). Разновидности протокола называются **диалектами**.

Microsoft и группа других вендоров (Digital Equipment, Data General, SCO, Network Appliance Corp, и т.д.) разработали открытую, официальную версию SMB протокола - CIFS 1.0. Он по существу тот же, что и NT LM 0.12, но с не большими доработками.

Простая модель SMB протокола

Элементы протокола (запросы и ответы), которыми обмениваются клиент и сервер, называются SMB. Они имеют определенный формат, который очень простой и для запросов, и для ответов. Каждый состоит из головной части фиксированного размера, параметров переменной длины и части данных.

8	16	24	32 bits
Command	RCLS	Reserved	ERR
ERR (cont)	REB/FLG	Reserved	
Reserved			
Reserved			
Reserved			
Tree ID		Process ID	
User ID		Multiplex ID	
WCT	VWV		
BCC		BUF	

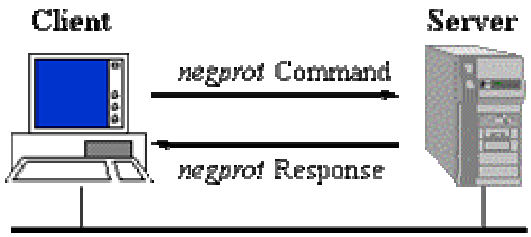
SMB header structure

Основные элементы структуры заголовка SMB:

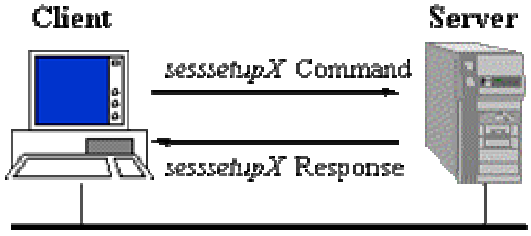
Command – команда протокола. **RCLS** – код класса ошибки. **ERR** – код ошибки. **Tree ID (TID)** – идентификатор соединения с сетевым ресурсом. **Process ID (PID)** – идентификатор клиентского процесса фактического соединения. **User ID (UID)** – идентификатор пользователя; используется сервером для проверки прав доступа пользователя. **Multiplex ID (MID)** - идентификатор группы пользователя; используется сервером для проверки прав доступа группы пользователя. **WCT** – количество параметров, следующих за заголовком. **BCC** – количество байт данных, следующих за параметрами.

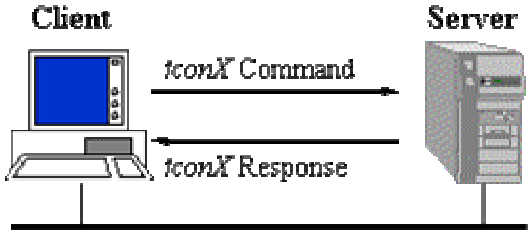
После соединения на NetBIOS уровне, либо посредством NBF, клиент готов делать запросы серверу. Однако клиент и сервер, для начала, должны определить, какой вариант протокола они оба понимают.

Клиент посылает **negprot (negotiate protocol)** SMB на сервер, перечисляя разновидности протокола, которые он понимает. Сервер, отвечает, передавая индекс варианта протокола, который он хочет использовать, или 0xFFFF, если ни один из вариантов не подходит. Диалекты, более поздние, чем Core и CorePlus, поддерживают передачу в negprot ответе информацию, описывающую их возможности.



Пусть теперь соединение установлено. Клиент аутентифицируется на сервере. Для этого клиент, должен послать SMB **sessetupX (session setup)**. Сервер проверяет имя пользователя и пароль. Если они валидны, то сервер обеспечивает клиента дополнительной информацией. Один из главных аспектов ответа sessetupX – это **UID** пользователя. UID должен всегда подтверждаться со всеми последующими SMB в текущем соединении. Теперь клиент залогинен. Посылается **tcon (tree connection)** или **tconX** SMB, определяющий ресурс в сети, к которому хочет обратиться клиент. Если необходимый доступ есть, то сервер отвечает, посылая клиенту **TID**, который клиент будет использовать все оставшееся время





использования этого ресурса. После соединения с ресурсом, клиент может посылать различные SMB, например, для чтения, записи и т.д.

Microsoft предложила общедоступную, открытую реализацию SMB протокола, который называется **CIFS (Common Internet File System).** Этот новый протокол более мощный и более гибкий, чем более ранние диалекты.

Аутентификация Microsoft SMB Protocol

Модель механизма защиты, которая используется в Microsoft SMB Protocol, в основном идентична модели любого другого варианта SMB протокола. Она состоит из двух уровней защиты: **user-level (пользовательский уровень)** и **share-level (уровень совместно используемого ресурса)**. Под **share (выложенный в сеть ресурс)** понимается файл, директория принтер, любая услуга, которая может быть доступна клиентам по сети.

Аутентификация на уровне **user-level** означает, что клиент, который пытается получить доступ к ресурсу на сервере, должен иметь **username (имя пользователя)** и **password (пароль)**. Если аутентификация прошла успешно, клиент имеет доступ ко всем доступным ресурсам сервера, кроме тех, что с **share-level** защитой. Этот уровень защиты дает возможность системным администраторам конкретно указывать, какие пользователи и группы пользователей имею доступ к определенным данным. Он используется в Windows NT, Windows 2000, Windows XP.

Аутентификация на уровне **share-level** означает, что доступ к ресурсу контролируется паролем, установленным конкретно на этот ресурс. В отличие от **user-level**, этот уровень защиты не требует имя пользователя для аутентификации и не устанавливается никакая уникальность текущего пользователя. Этот уровень используется в Windows NT, Windows 2000 и Windows XP для обеспечения дополнительного уровня контроля защиты сверх **user-level**. Операционные системы Windows 95, Windows 98 и Windows ME реализуют защиту только этого уровня.

В обоих этих уровнях защиты используется шифрование. Пароль зашифровывается, прежде чем отправляется на сервер. Типы шифрования NTLM и старые версии LAN Manager (LM) поддерживаются протоколом. Оба метода шифрования используют аутентификацию типа *отклик-отзыв*, в которой сервер посылает клиенту случайную сгенерированную строку, а клиент возвращает в качестве отзыва обработанную строку, которая доказывает, что клиент имеет достаточный мандат для доступа к данным.

Microsoft NTLM (Windows NT LAN Manager) – это протокол аутентификации, который используется в сетях, имеющих рабочие станции с Windows NT. Мандат NTLM базируется на данных полученных в течение процесса интерактивного входа в систему и состоит из имени пользователя, имени домена и **хеша (результат применения криптоалгоритма над произвольным количеством данных, имеющий фиксированную длину)** пароля пользователя. NTLM использует модель *отклик-отзыв* для аутентификации пользователя без передачи пароля через сеть.

NTLM аутентификация происходит следующим образом:

1. Пользователь предоставляет следующие данные (username, password, domain name). Клиент вычисляет значение криптографической хеш функции пароля и отбрасывает реальный пароль.
2. Клиент посылает username на сервер в виде простого незашифрованного текста.
3. Сервер генерирует случайное 16-битное число, которое называется *откликом.*
4. Клиент принимает отклик, шифрует его хэшем пароля пользователя и отправляет результат на сервер. Это называется *отзывом*.
5. Сервер посылает username, *отклик* и *отзыв* в контроллер домена.
6. Контроллер домена использует username для получения хэша пароля пользователя из своей базы данных **SAM (Security Account Manager)**. Затем, с помощью этого хэша расшифровывается *отклик*.

7. Контроллер домена сравнивает расшифрованный *отклик* и *отзыв*. Если они идентичны, то аутентификация проходит успешно.

Для вычисления хеша пороля пользователя строка пароля конвертируется в бинарную форму. Для этого может применяться два криптоалгоритма в зависимости от того, следует ли протоколу учитывать регистр пароля.

В криптоалгоритме, чувствительному к регистру, применяется алгоритм хеширования **MD4**. 128-битный результат применения этого алгоритма как раз и выступает в роли хеша пароля.

Сложнее обстоит дело, когда регистр не учитывается. Сначала строка пароля преобразуется в верхний регистр. **”Password” -> “PASSWORD”**. Если длина пароля меньше 14 символов, то недостающие символы дополняются нулями. Затем с помощью, **DES** шифруется строка **“KGS!@#$%”**. Происходит это в два этапа. Первые 7 байт пароля дают первый ключ, вторые - второй, конкатенация которых дает нам искомый 128-битный хэш пароля пользователя, необходимый для аутентификации:

key1 = DES(“KGS!@#$%”, Password[0..6])
key2 = DES(“KGS!@#$%”, Password[7..13])
Kl = key1 || key2.

Методы взлома SMB.

Большинство методов взлома основано на том, что злоумышленник получает доступ к базе данных, в которой операционная система хранит хеши паролей пользователей, т.е в SAM. Если SAM уже получена, то далее остается только воспользоваться одной их многих утилит, которые помогают дешифровать пароли пользователей, полученные из SAM. Одной из самых известных подобного рода программ является **L0phtCrack**. Принцип ее действия очень прост. Похищенная база данных сначала импортируется в программу. Далее выбирается нужная учетная запись и запускается процесс дешифровки, который может занять много времени. Среди способов расшифровки имеется возможность проведения “атаки по словарю”, т.е попытка угадать пароль по уже заложенному в программу списку распространенных паролей, на тот случай если пароль короткий и незамысловатый. Также злоумышленник может задумать использовать L0phtCrack в режиме "SMB Packet Capture" в локальной сети, если он обладает необходимыми правами для установки и запуска программ на компьютере. В этом случае хеши паролей пользователей также можно будет перехватить с помощью утилиты. Запустив программу сразу же можно, перехватывая пакеты, получить хеши паролей пользователей, которые аутентифицируются на сервере в данный момент, так как довольно часто при работе в локальной сети пользователи с разных компьютеров повторно аутентифицируются на главном сервере. Затем злоумышленнику остается только сохранить все перехваченные хэши паролей и начать дешифрацию аналогично тому, как это происходило с дешифровкой SAM файла. Как правило, у большинства пользователей простые пароли, поэтому программа способна взломать их всего за несколько минут своей работы, с помощью встроенного списка паролей. Остальные утилиты основаны на таком же принципе.

Литература

1. **“Just what is SMB?”,** Richard Sharpe, 08.10.2002. http://samba.anu.edu.au/cifs/docs/what-is-smb.html
2. **“Microsoft SMB Protocol and CIFS Protocol Overview”**, MSDN Library. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/fileio/base/microsoft_smb_protocol_and_cifs_protocol_overview.asp
3. “**SMB Protocol**”, Nikolay Kultashev, 30.11.1999. http://iroi.seu.edu.cn/books/ee_dic/whatis/smb.htm
4. **“Защита и безопасность в сетях LINUX”**, Дэвид Бэндл, Питер, 2002.
5. **“IBM Protocols: SMB”.** http://www.protocols.com/pbook/ibm.htm#SMB
6. **“IBM SMB: Server Message Block protocol”.** http://www.javvin.com/protocolSMB.html
7. **“Получение прав администратора в Windows NT”.** Евгений Сечко. http://wahack.org.ru/bintxt/hacknt1.shtm.