Шифр:
Кросс

Студенческая научная работа по профильной дисциплине:
«Информационная безопасность»
Тема: «Анализ сетевого протокола многопользовательской онлайн игры»

Содержание

Описание протокола авторизации

Процесс авторизации на login-сервере и подключение к game-сервер.

Авторизация на логин-сервере происходит в несколько этапов.

1) логин-сервер отправляет пакет инициализации длиной 11 байт (содержит информацию о версии).

0B 00 00 74 44 DE 5B 5A 78 00 00

где байты 5A 78 информируют о том, что версия протокола сервера 785A

2) клиент отправляет пакет RequestAuthLogin с запросом содержащий логин и пароль. Ниже приведено его описание на примере:

Таблица 1 – RequestAuthLogin

32 00	длина пакета
00	тип пакета
61 6D 35 31 00 00 00 00 00 00 00 00 00 00	строка, содержащая логин. Имеет длину 14 байт, храниться в формате ASCII. Логин: am51
61 6D 35 31 00 00 00 00 00 00 00 00 00 00 00 00	строка, содержащая пароль. Имеет длину 16 байт, храниться в формате ASCII. Пароль: am51
08	маркер конца секции логин/пароль
00 00 00 00	не используется
00 00 00 00	не используется
5C 54 5C 5C 00 00 00 00	контрольная сумма

3) если пароль верный, сервер высылает пакет LoginOk с 32-битным номером нашего профиля – SessionKey #1. Ниже приведено его описание на примере:

Таблица 2 – LoginOk

32 00	длина пакета
03	тип пакета
04 00 00 00	SessionKey #1 часть 1
D3 2C 18 13	SessionKey #1 часть 2

00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	константа
10 D6 2C 18 00 00 00 00	контрольная сумма

4) клиент отправляет пакет RequestServerList, на что сервер отвечает списком серверов, содержащим ip адреса, порты, число online пользователей, максимальное число пользователей.

5) клиент отправляет пакет RequestServerLogin. Сервер проверяет AccessLevel профиля (если он равен -1 профиль заблокирован на сервере) и в зависимости от логина, пароля, уровня доступа и сокета, генерирует уникальный 32-битный SessionKey #2, по которому в последствие авторизует game-сервер.

Если же игровой сервер в состоянии down, или имитирует это состояние (администрация делает это для проведения профилактических работ на сервере) или число online пользователей достигло максимума, высылается пакет с причиной невозможности авторизации.

SessionKey #2 находится в пакете PlayOk. Ниже приведено его описание на примере:

Таблица 3 – PlayOk

1A 00	длина пакета
07	тип пакета
0A 00 00 00	SessionKey #2 часть 1
04 00 00 00	SessionKey #2 часть 2
01 00 00 00 00 00 00	константа
07 0F 00 00 00 00 00 00	контрольная сумма

6) Если все этапы пройдены успешно, подключение на игровой сервер возможно, для этого необходимо отправить игровому серверу пакет инициализации (для каждого сервера он свой, но константный), на что он отвечает 12-ти байтным пакетом Init, содержащим первые 4 байта ключа, которые скрепляются с другими 4-мя байтами (которые постоянны), в

результате будет получен 64-битный ключ. В дальнейшем этот ключ будет использоваться для расшифровки и зашифровки игровых пакетов. Важно отметить, что с каждым рас(за)шифрованным пакетом, его длина прибавляется к первой части ключа.

7) клиент отправляет пакет AuthRequest содержащий логин и два идентификатора (уже в зашифрованном виде), которые были получены в сеансе с login-сервером. В ответ от сервера приходит пакет со списком персонажей. Ниже приведено описание AuthRequest на примере:

Таблица 4 – AuthRequest

21 00	длина пакета
08	тип пакета
61 00 6D 00 35 00 31 00 00 00	строка, содержащая логин. Храниться в формате Unicode. Логин: am51
04 00 00 00	SessionKey #2 часть 2
0A 00 00 00	SessionKey #2 часть 1
04 00 00 00	SessionKey #2 часть 2
D3 2C 18 13	SessionKey #1 часть 2
04 00 00 00	SessionKey #1 часть 1

Стоит отметить, что в новых версиях клиента и сервера произошла смена протокола. Добавилось шифрование алгоритмом RSA и ключ Blowfish стал динамическим [17]. Рассмотрим новый вариант авторизации:

1) Сразу после установки соединения сервер отправляет клиенту пакет Init длинной 186 байт. Содержит новый ключ Blowfish и открытый ключ RSA [18]. Ниже приведено его описание на примере:

Таблица 5 – Init

BA 00	длина пакета
00	тип пакета
68 01 00 00	байты необходимые для пакета RequestGGAuth
21 C6 00 00	версия протокола сервера C621

F9 A7 44 CD 09 4D 64 78 11 C7 07 36 35 F1 EA 8B 75 66 C4 B3 CD F4 C9 4B D5 2F 8E 70 A8 30 CE 3E 51 25 A9 61 B1 D7 B1 3E 0D ED 9C 2D 80 41 19 F8 EF BE 84 03 48 D7 EC 96 4E D7 36 EC 8F AE ED A9 FB B7 40 47 FF 79 B3 68 7C 29 CD BC 91 17 7D 80 49 26 AD 9F BC FB C1 A0 FD 98 2C 56 31 0E AC 7C F8 69 62 30 81 34 A5 82 4A 9B 8F FF 52 94 A0 00 FF A8 DC A5 96 AD E8 62 46 06 F4 F4 98 90 F5 8B	открытый RSA ключ
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	константа
3F 6B 68 09 6C 0B EF 6E B6 4F A9 C1 B6 4F A9 C1	новый ключ Blowfish
00 00 00 00 00 00 00	константа
74 AC E8 BB	контрольная сумма
9A AB 66 C9	XOR ключ которым зашифрован был пакет

2) В ответ на него клиент отправляет пакет RequestGGAuth.

Ниже приведено его описание на примере:

Таблица 6 – RequestGGAuth

2A 00	длина пакета
07	тип пакета
68 01 00 00	байты, полученные с пакета Init
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	константа
07	тип пакета
68 01 00 00	байты, полученные с пакета Init

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	константа

3) Сервер отвечает на него пакетом GGAuth.

4) Если сервер ответил, что авторизация прошла успешно, то клиент высылает пакет RequestAuthLogin, содержащий логин и пароль. Ниже приведено его описание на примере:

Таблица 7 – RequestAuthLogin

B2 00	длина пакета
00	тип пакета
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	константа. Часть блока хранящего логин и пароль
24 00 00	константа. Часть блока хранящего логин и пароль
61 6D 35 31 00 00 00 00 00 00 00 00 00 00	строка, содержащая логин. Имеет длину 14 байт, храниться в формате ASCII. Логин: am51
61 6D 35 31 00 00 00 00 00 00 00 00 00 00 00 00	строка, содержащая пароль. Имеет длину 16 байт, храниться в формате ASCII. Пароль: am51
00 00 00 00	константа. Часть блока хранящего логин и пароль
68 01 00 00	байты, полученные с пакета Init
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	константа

08	константа
00 00 00 00 00 00 00 00 00 00	константа
F5 18 6A 0D 00 00 00 00	контрольная сумма
00 00 00 00 00 00 00 00	константа

5) Проверка логина и пароля, в случае неудачи, сервер высылает пакет LoginFail содержащий причину неудачи, иначе высылается пакет LoginOk, содержащий session key #1.

Пакет Init сначала шифруется по алгоритму XOR(ключ генерируется случайным образом и помещается в конце пакета, код алгоритма XOR приведен в приложении), а потом шифруется по алгоритму Blowfish, статическим ключом. По умолчанию статический ключ – 6B 60 CB 5B 82 CE 90 B1 CC 2B 6C 55 6C 6C 6C 6C. Все последующие пакеты будут шифроваться динамическим Blowfish ключом. Часть пакета RequestAuthLogin дополнительно шифруется по алгоритму RSA. Ключ состоит из следующих частей: B = 1024, E = 65537, N = передается в пакете Init. Вместе эти 3 части составляют целый RSA ключ.

Выводы

Если игра класса MMORPG популярна, то в большинстве случаев найдутся альтернативы официальным серверам в виде бесплатных, открытых (или платных и закрытых) так называемых проектов. Lineage популярна и она имеет альтернативный сервер, отличный от официального сервера, но всячески стремящийся стать похожим на официальный сервер. Существует достаточное множество альтернативных клиентов, так называемых «ботов». Называются они так, потому что они могут вместо игрока исполнять основные игровые действия: повышение уровня игрока, продажа игровых вещей и т.д. На большинстве серверов они запрещены, так как это даёт нечестным игрокам получить преимущество. Но не существует ни единого упоминания о портативных версиях этой игры. Lineage2 очень популярная игра, и даже на простой графике портативных устройств люди захотят играть.

В данной работе выполнен анализ дампа обмена данными клиент-серверного приложения, показано, что защита не выполнена в должной мере и разработан клиент для подключения к официальным серверам. При этом он работает как на портативных устройствах, так и на компьютере, и особых различий в клиенте для разных платформ нет. Частично этот функционал покрывают «боты» программы, но только в рамках компьютера, и у них другая задача – как клиент они не позиционируются. Как сама идея создания игры на мобильных устройствах, возможно, не нова, но для мира Lineage – это новое и попыток воссоздания клиента не было. Для написания собственного клиента игры, необходимо провести дальнейший анализ защищенного протокола передачи данных.

Список литературы

1. TCP/IP - Википедия. Википедия – свободная энциклопедия. [В Интернете] http://ru.wikipedia.org/wiki/TCP/IP.

2. Олифер В.Г., Олифер Н. А. Компьютерные сети. Принципы, технологии, протоколы. – СПб. : Питер, 2001. – 672 с.

3. Microsoft. Общие сведения об основных понятиях платформы .NET Framework. Веб-узел корпорации Microsoft. [В Интернете] http://msdn.microsoft.com/ru-ru/library/zw4w595w(v=VS.90).aspx.

4. Microsoft. Общие сведения о системе общих типов (CTS). Веб-узел корпорации Microsoft. [В Интернете] http://msdn.microsoft.com/ru-ru/library/2hf02550(v=vs.90).aspx.

5. Microsoft. Общие сведения о взаимодействии кодов на разных языках. Веб-узел корпорации Microsoft. [В Интернете] http://msdn.microsoft.com/ru-ru/library/a2c7tshk(v=vs.90).aspx.

6. Microsoft. Обзор графических возможностей. Веб-узел корпорации Microsoft. [В Интернете] http://msdn.microsoft.com/ru-ru/library/d0ezbwf0(v=vs.90).aspx.

7. Microsoft. Три категории графических служб. Веб-узел корпорации Microsoft. [В Интернете] http://msdn.microsoft.com/ru-ru/library/zccx11ha(v=vs.90).aspx.

8. Маклин С., Нафтел Дж, Уильяме К. Microsoft .NET Remoting. – М. : Издательско-торговый дом «Русская Редакция», 2003. – 384 с.

9. Microsoft. Socket - класс. Веб узел корпорации Microsoft. [В Интернете] http://msdn.microsoft.com/ru-ru/library/system.net.sockets.socket.aspx.

10. Microsoft. TcpClient - класс. Веб-узел корпорации Microsoft. [В Интернете] http://msdn.microsoft.com/ru-ru/library/system.net.sockets.tcpclient.aspx.

11. Microsoft. NetworkStream - класс. Веб-узел корпорации Microsoft. [В Интернете] http://msdn.microsoft.com/ru-ru/library/system.net.sockets.networkstream.aspx.

12. Schneier B. Schneier on Security. [В Интернете] http://www.schneier.com/.

13. Панасенко С. Интересные алгоритмы шифрования. [В Интернете] http://www.bytemag.ru/articles/detail.php?ID=9090&phrase_id=88593.

14. RSA - Викиедия. Википедия – свободная энциклопедия. [В Интернете] http://ru.wikipedia.org/wiki/RSA.

15. Menezes A., P. van Oorschot, Vanstone S. Handbook of Applied Cryptography. б.м. : CRC-Press, 1996. — 816 с .

16. Boneh D. Twenty years of attacks on the RSA cryptosystem. Notices of the American Mathematical Society. 1999 г.

17. L2j Server Project. L2j Server. [В Интернете] http://www.l2jserver.com/.

18. TechnoWiz@rd. Lineage II Packets. Протокол Lineage II. [В Интернете] http://fursoffers.narod.ru/Packets.htm.