RRC

Результаты тестирования шлюза x2500 под управлением м150

Тестирование провёл: Pete Kuzeev, Security Presale Engineer, RRC Moscow

Благодарности: Thomas Drews - Sr. Systems Engineer
Barry Shteiman, Johan Nordstrom, Alexander Kolybelnikov

RRC

Основные цели тестирования

Поскольку основной акцент при использовании системы делается на защиту СУБД, то в первую очередь были отмечены самые популярные из них на сегодняшний день – **MS SQL, Oracle, IBM DB2 и MySQL**

Одним из наиболее критичных приложений для деятельности организаций, работающих с данными СУБД, является **1С** – поэтому часть тестов пришлась и данное ПО

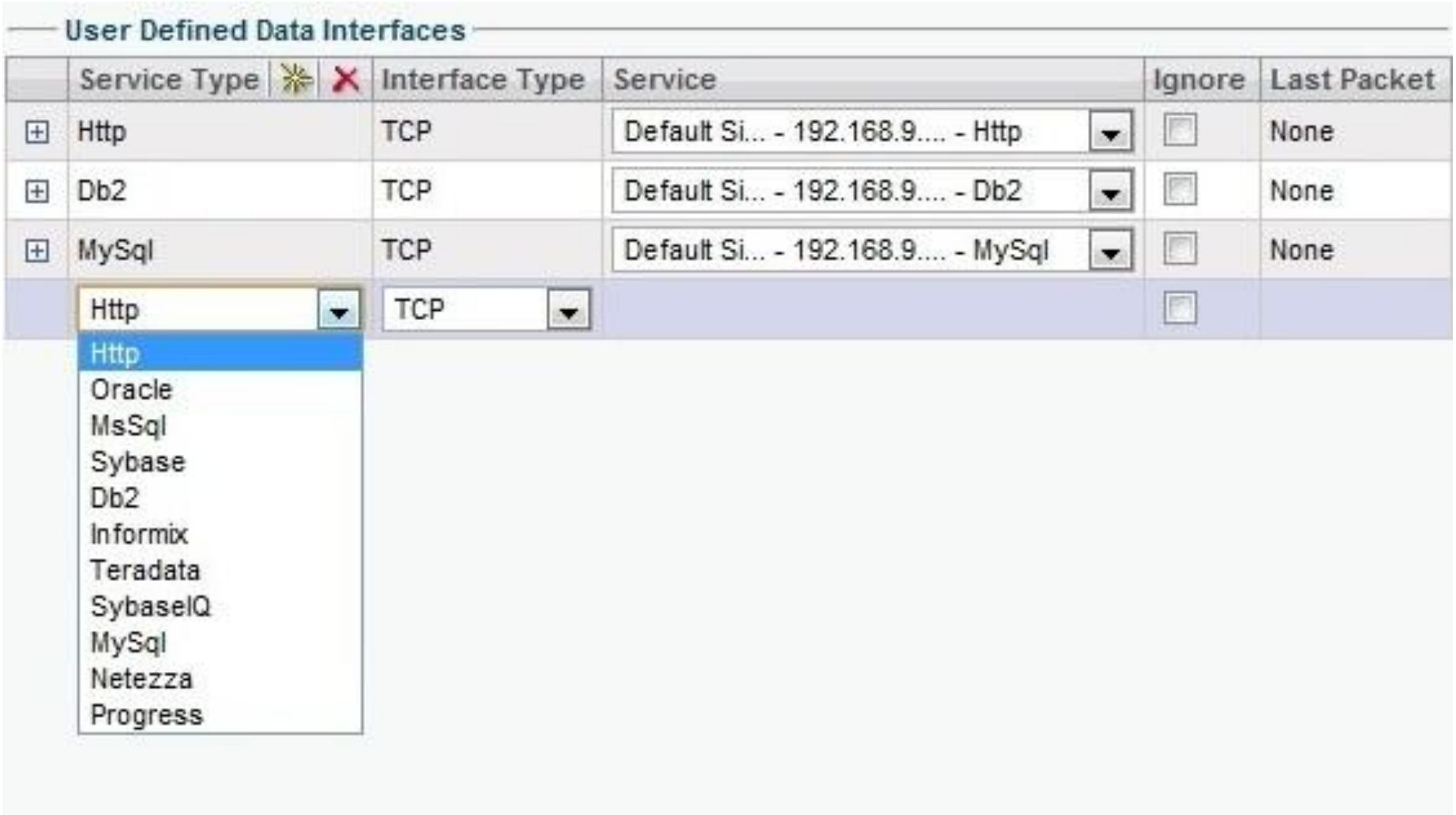
Многие использует порталы для внешних и внутренних корпоративных ресурсов, даже у 1С есть веб-сервисы, поэтому мы также добавили на стенд самую популярную связку **Apache + PHP + MySQL**

Традиционно весьма уязвимая, но работающая на многих системах без ведома администраторов MS IIS в связке **Web + Ftp**

Системы автоматизированного проектирования уже давно обрели статус систем способных «освоить» любые мощности и период, когда для хранения своих данных они довольствовались локальными таблицами и библиотеками в прошлом – **SAP + Oracle**

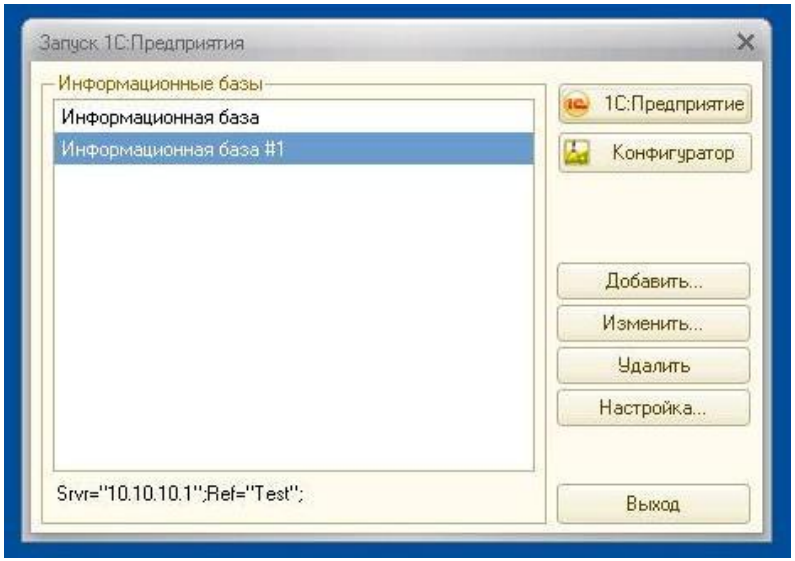
RRC

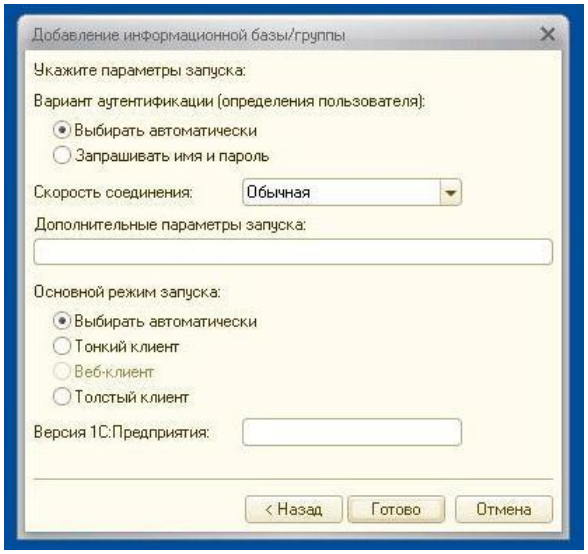
Список «любимых» СУБД

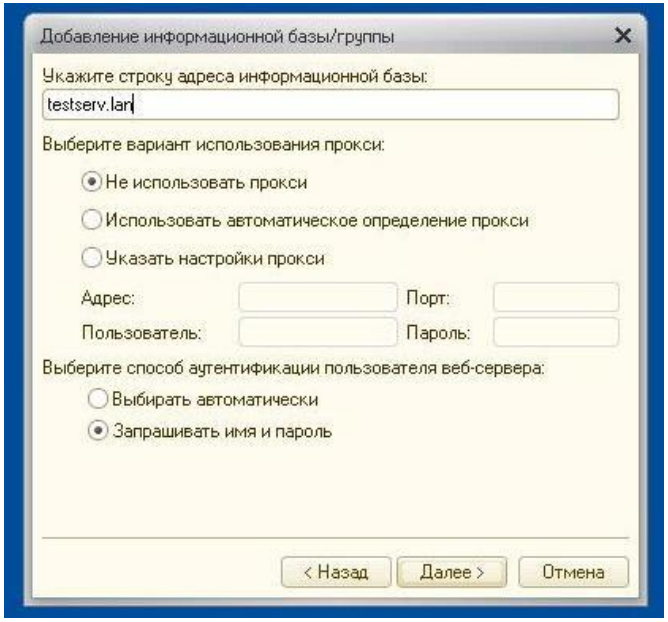


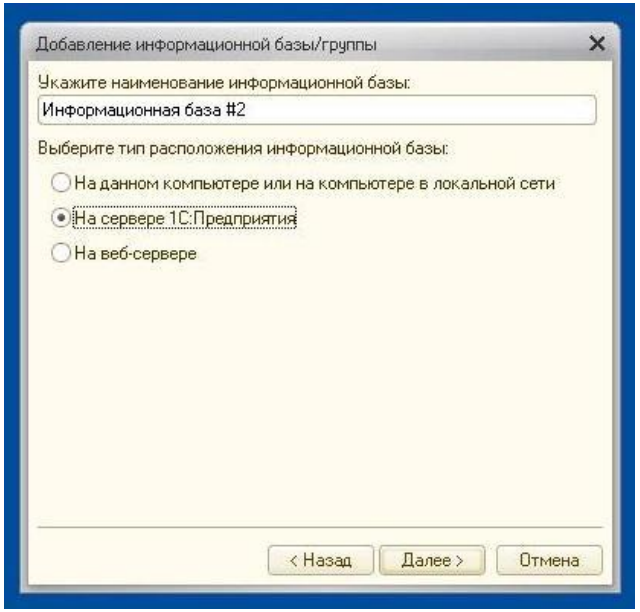
RRC

«Облачная» 1С

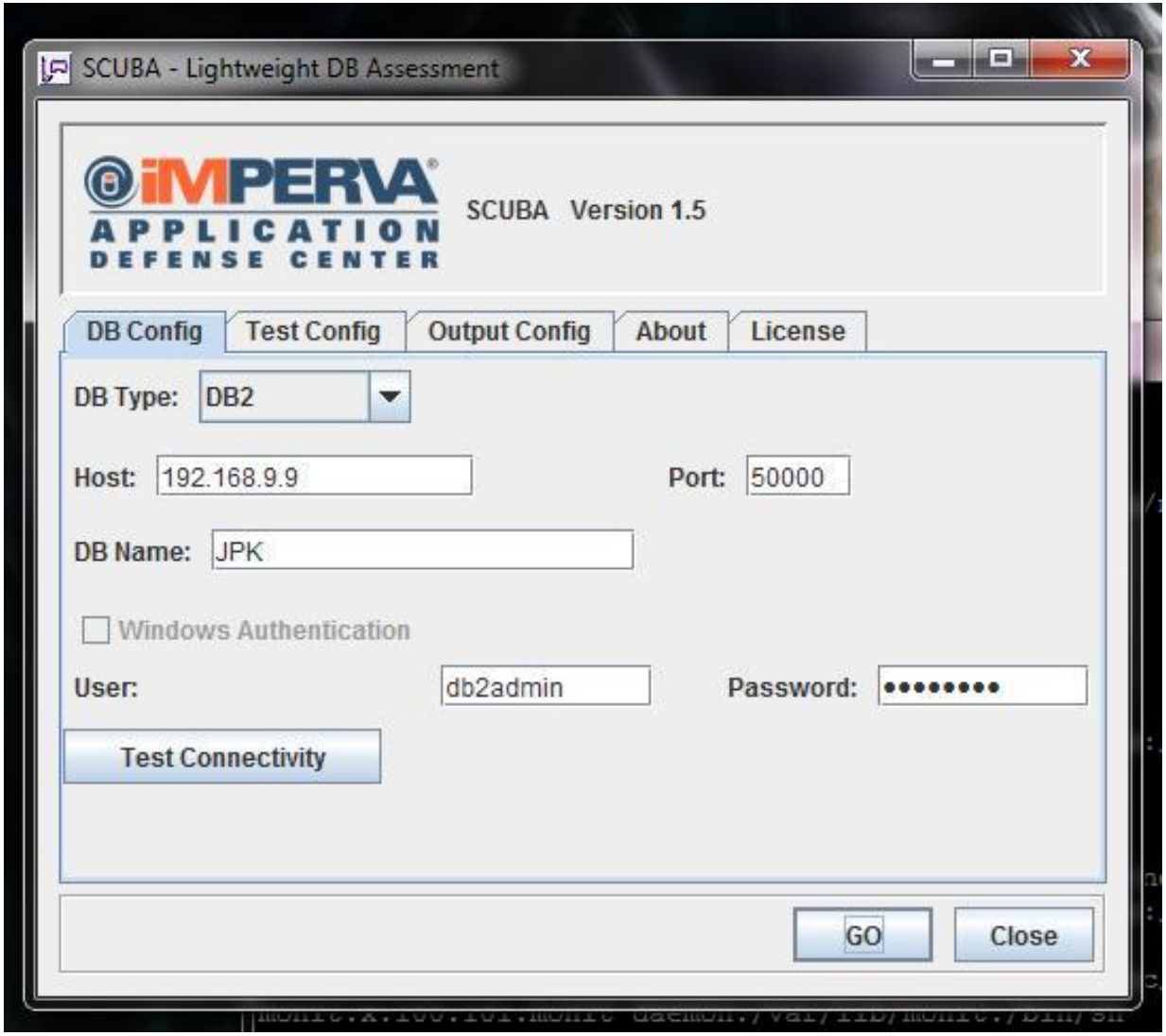








SCUBA – Imperva ADF



Инструменты для тестов

SCUBA – сканер уязвимостей СУБД от Imperva

Retina, X-Spider, Nessus, Acunetix – сканеры уязвимостей

NeXpose – сканер уязвимостей с динамикой (flood)

SQL Stripes – система мониторинга

ERPSsan – сканер уязвимостей SAP

Помимо всего выше перечисленного, позволяющего получить некоторые данные в автоматизированном режиме, был использован ряд утилит и множество скриптов на Perl написанных под конкретные задачи или популярные уязвимости…

Список тестируемого ПО – MS SQL 2008/2005, Oracle 11g Express, IBM DB2 9.7.4 Express, Apache 2.1, PHP 5.0, MySQL 5.1, 1C 8.2, SAP R3 4.6D r8

Конфигурация стенда

Стенд состоял из двух устройств от Imperva – шлюз x2500 и сервер управления m150, использовалось три подсети – одна для управления сервером, вторая для работы сервера и шлюза и третья для защищаемых сервисов

Хочу обратить ваше внимание на то, что мы использовали шлюз в режиме прозрачного моста, т.е. рассмотрен самый оптимальный вариант, когда для внедрения устройства не требуется вносить никаких изменения в рабочую сетевую инфраструктуру

Сервисы были установлены как реальных, так и на виртуальных машинах, особой разницы в тестах, кроме ожидаемых временных задержек мы не обнаружили

Работа комплекса

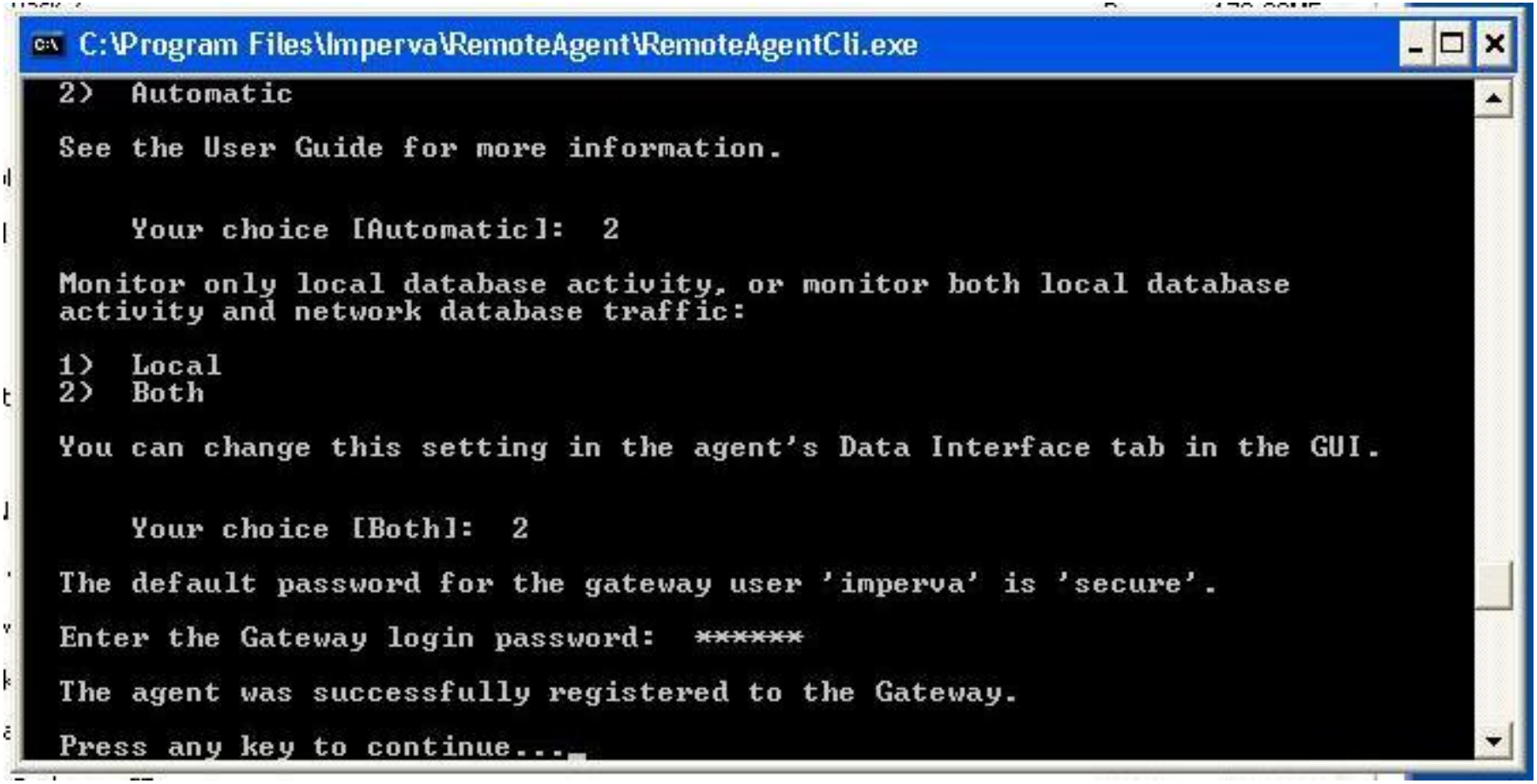
По умолчанию работа комплекса ограничивается установкой и настройкой самих устройств без нагрузки защищаемых клиентов каким-либо дополнительным ПО, но существуют ситуации, при которых, установка клиента (в нашем случае – Imperva Agent) является единственным вариантом для получения тех или иных данных , например – загрузка ЦП

Imperva Agent – решение клиента в виде сервиса с весьма скромными требованиями к целевой системе, его присутствие в ней не замедляет работу и не затрудняет взаимодействие развёрнутых сервисов

Администрировать комплекс можно через защищённый веб-интерфейс, который доступен через специальные порты управления. Стандартный доступ по SSH, а также отдельный консольный порт, поддерживающие режимы native unix - для основных системных настроек

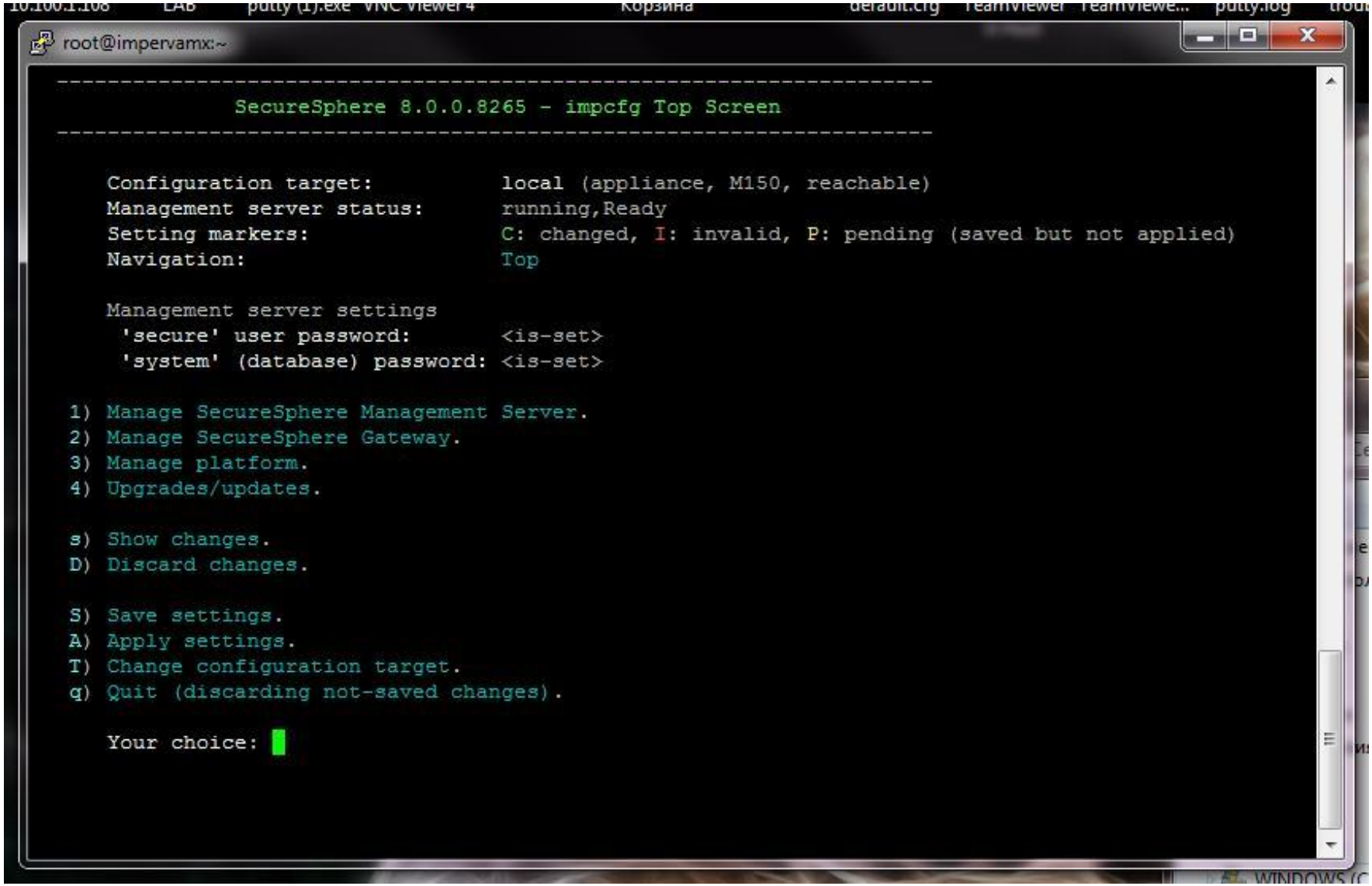
RRC

Установка агента



RRC

Интерфейс Impcfg



Правила и политики

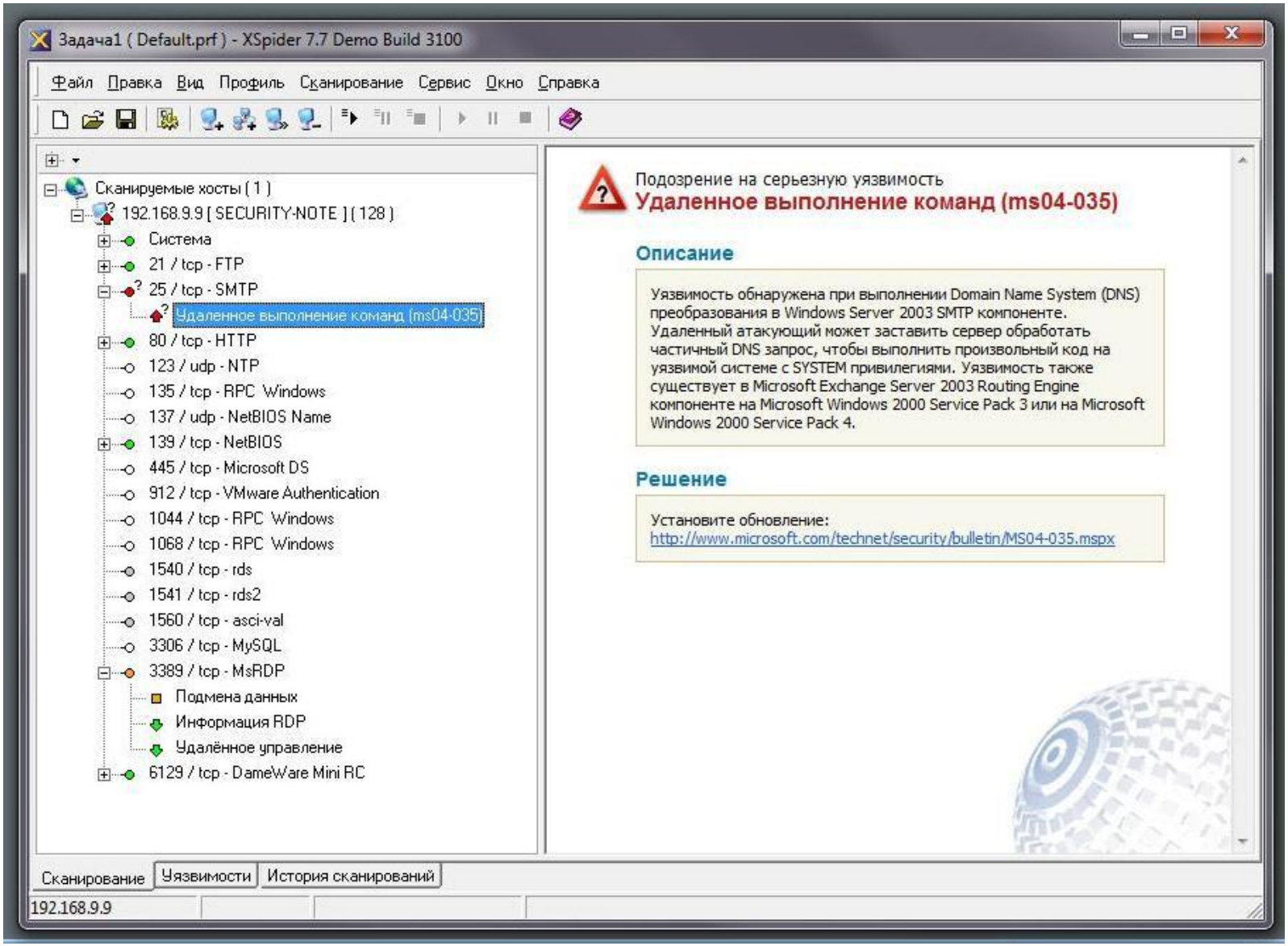
Когда на в руки попадает «красивая вещь» – первым делом приходит мысль о такой полезной функции, как конфигурация по умолчанию. Хочется распаковать апплайнс и проведя начальное конфигурирование получить более менее рабочую систему, так мы и поступим

Собрали и настроили стенд по описанию, данному ранее, и запустили предварительное сканирование тестовой защищаемой машинки из двух подсетей – внешней (подразумеваем интернет) и подсети управления, через которую мы тоже можем постучать на защищаемую машинку

На следующем слайде приведены результаты сканирования одной из программ, если заметить в целом, то первое впечатление о политиках безопасности по умолчанию – очень хорошее, из подсети управления, которая считается доверенной, все соединения по поиску средних и высоких по опасности уязвимостей разрешаются это видно в сканере и апплайнс просто информирует нас, о таких попытках, из общественной сети – все данные соединения отсеиваются шлюзом, без каких-либо дополнительных настроек

RRC

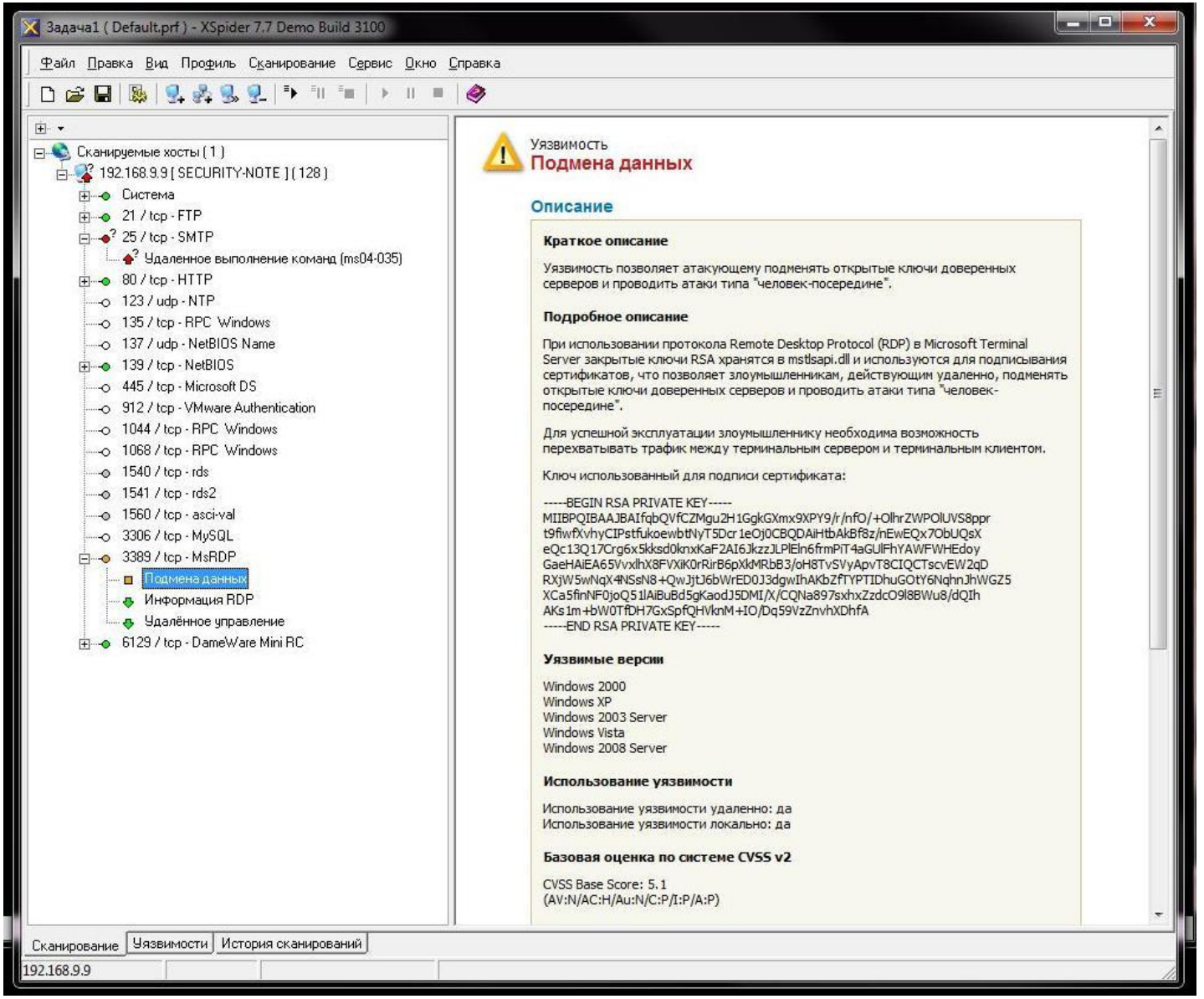
Работа шлюза по умолчанию



Удачные попытки сканирования из доверенной сети

RRC

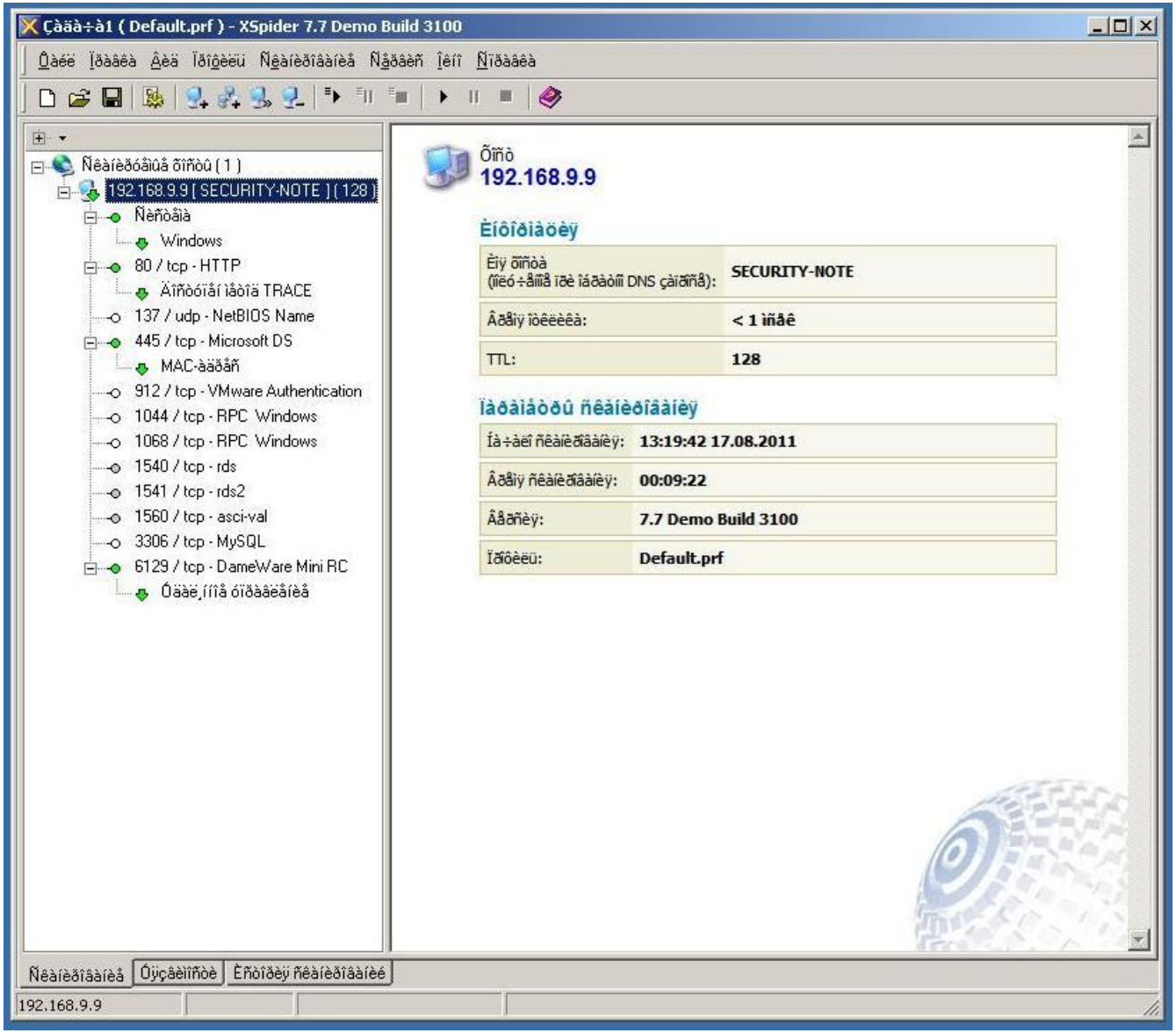
Работа шлюза по умолчанию



Удачные попытки сканирования из доверенной сети

RRC

Работа шлюза по умолчанию



Попытки сканирования из небезопасной сети – критичные уязвимости отфильтрованы

Тесты СУБД

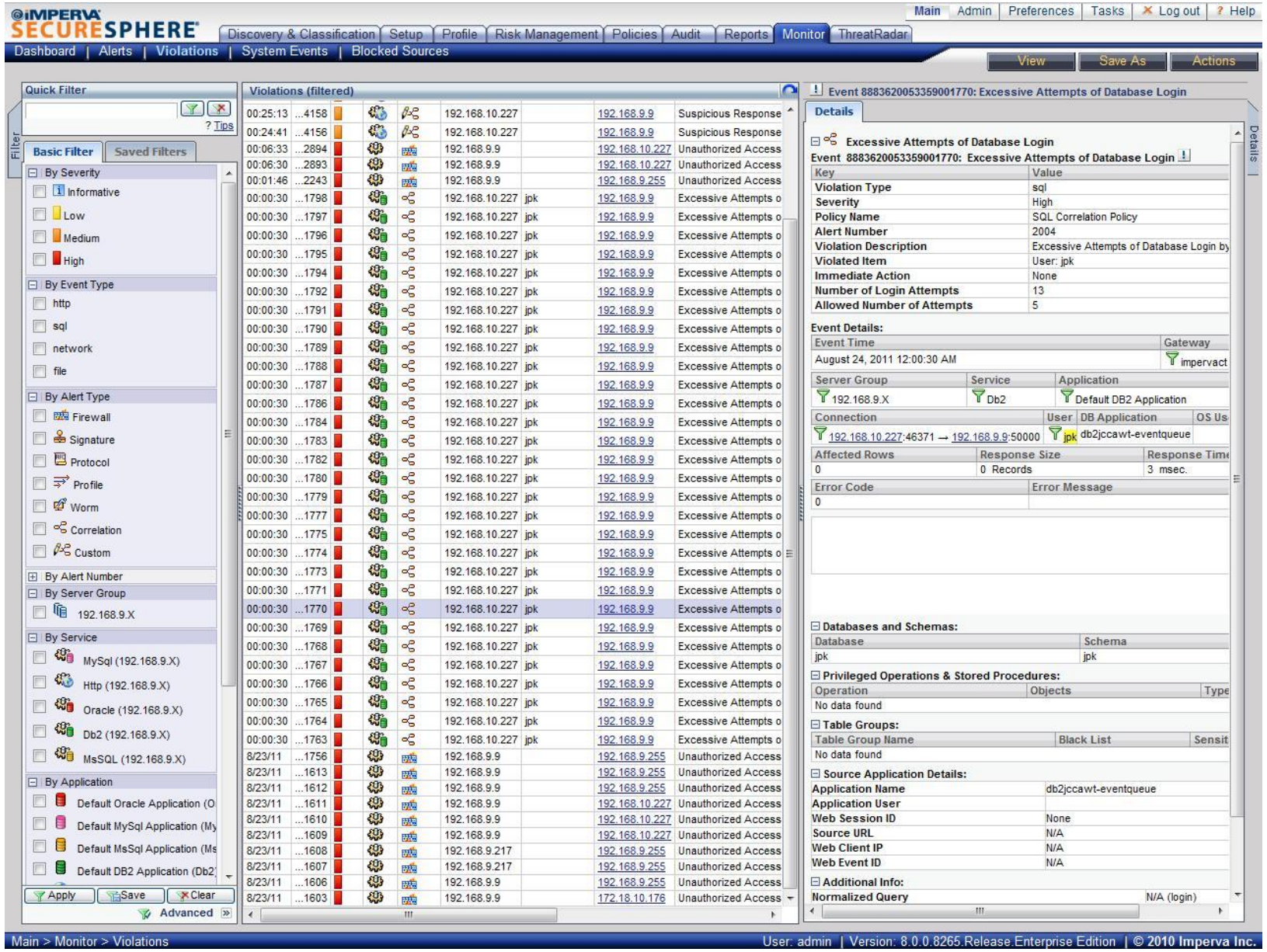
Поскольку основное направление нашего тестирования СУБД – вторым пунктом в наших тестах стояли именно они, маленькая ремарка, для тестов были применены политики предлагаемые системой как оптимальные для защиты наших сервисов, т.е. мне не нужно было тратить большое количество времени на оставление карты потенциальных уязвимостей

Хочу отметить самые важные тесты в области защиты СУБД - это создание запроса, позволяющего извлечь любые системные данные, обойти проверку пользователя при доступ к определённым таблицам, а также для некоторых СУБД (например, Oracle) получить доступ к файловой системе и произвести загрузку некоторого исполняемого кода, для получения полного доступа к системе

Тестировались как БД созданные нами, так и БД созданные 1С и SAP R3 – в обоих случаях ни один из целевых запросов, эксплуатирующих критичные уязвимости не вошёл за границы защищаемого шлюзом Imperva периметра, на наш взгляд - **это отличная перспектива забыть о постоянных обновлениях и патчах для СУБД, которые порой снижают её производительность, а также получить великолепную систему мониторинга и оповещения**

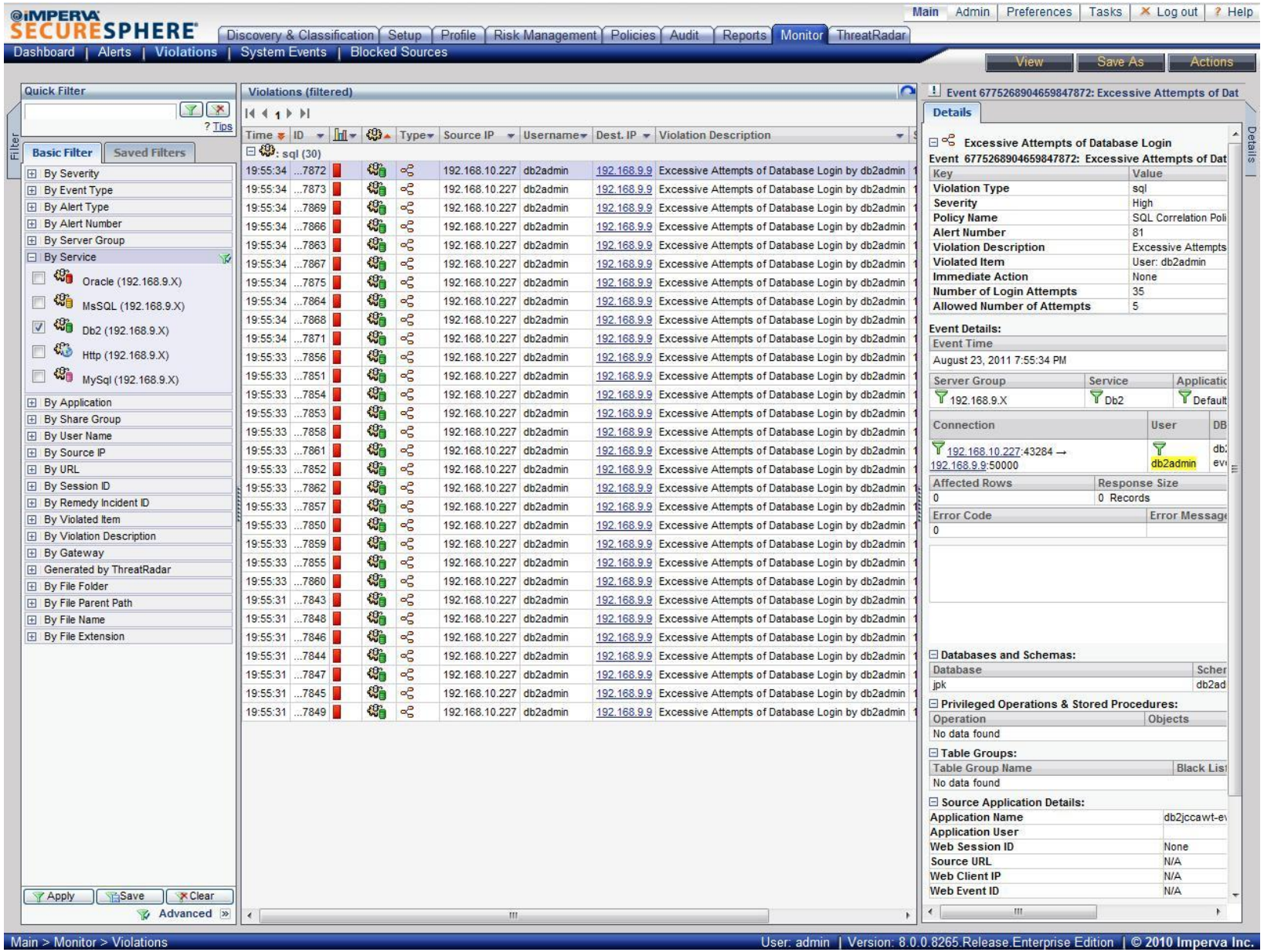
RRC

Тесты СУБД – Монитор/Список



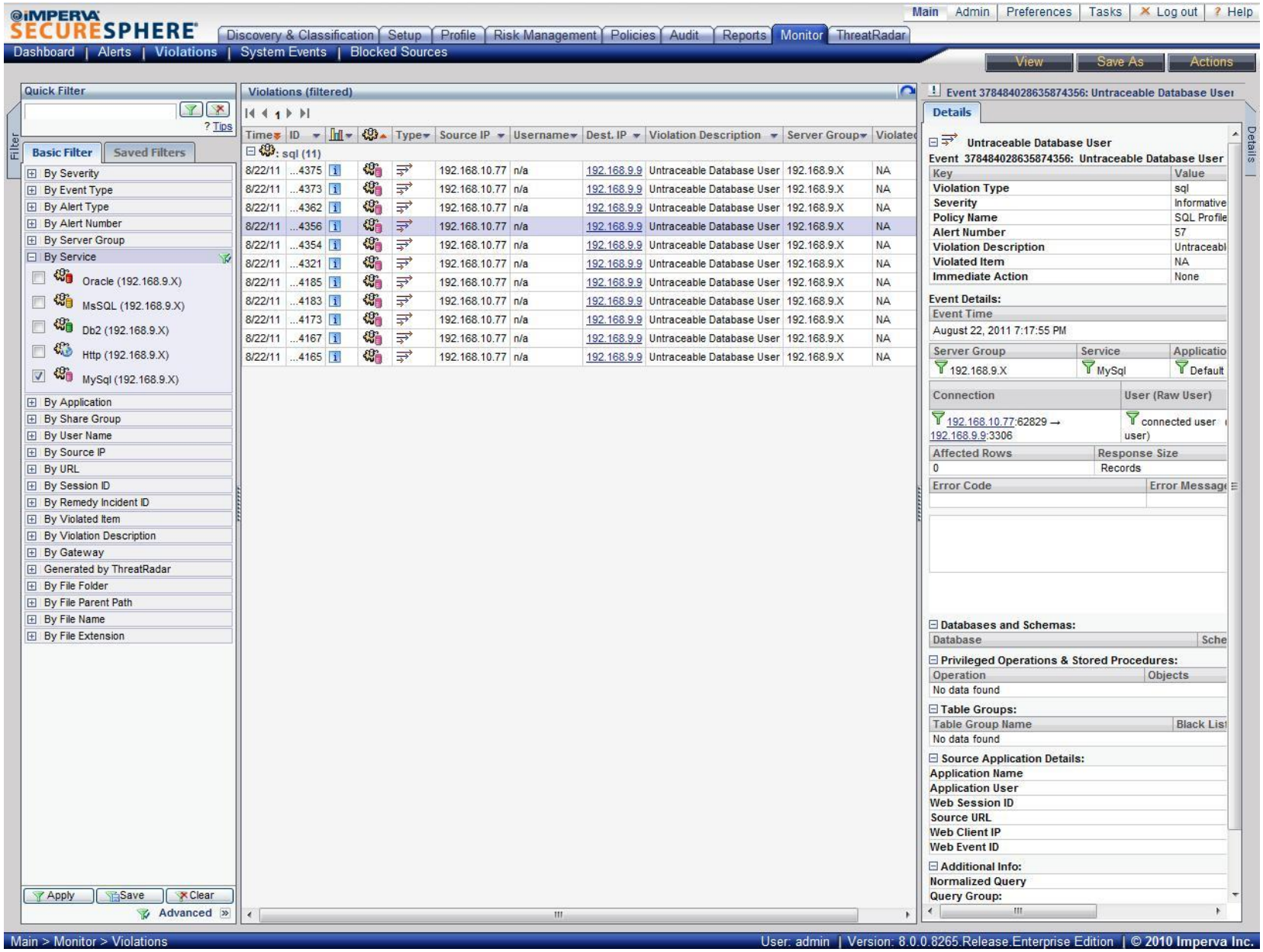
RRC

Тесты СУБД – Монитор/DB2



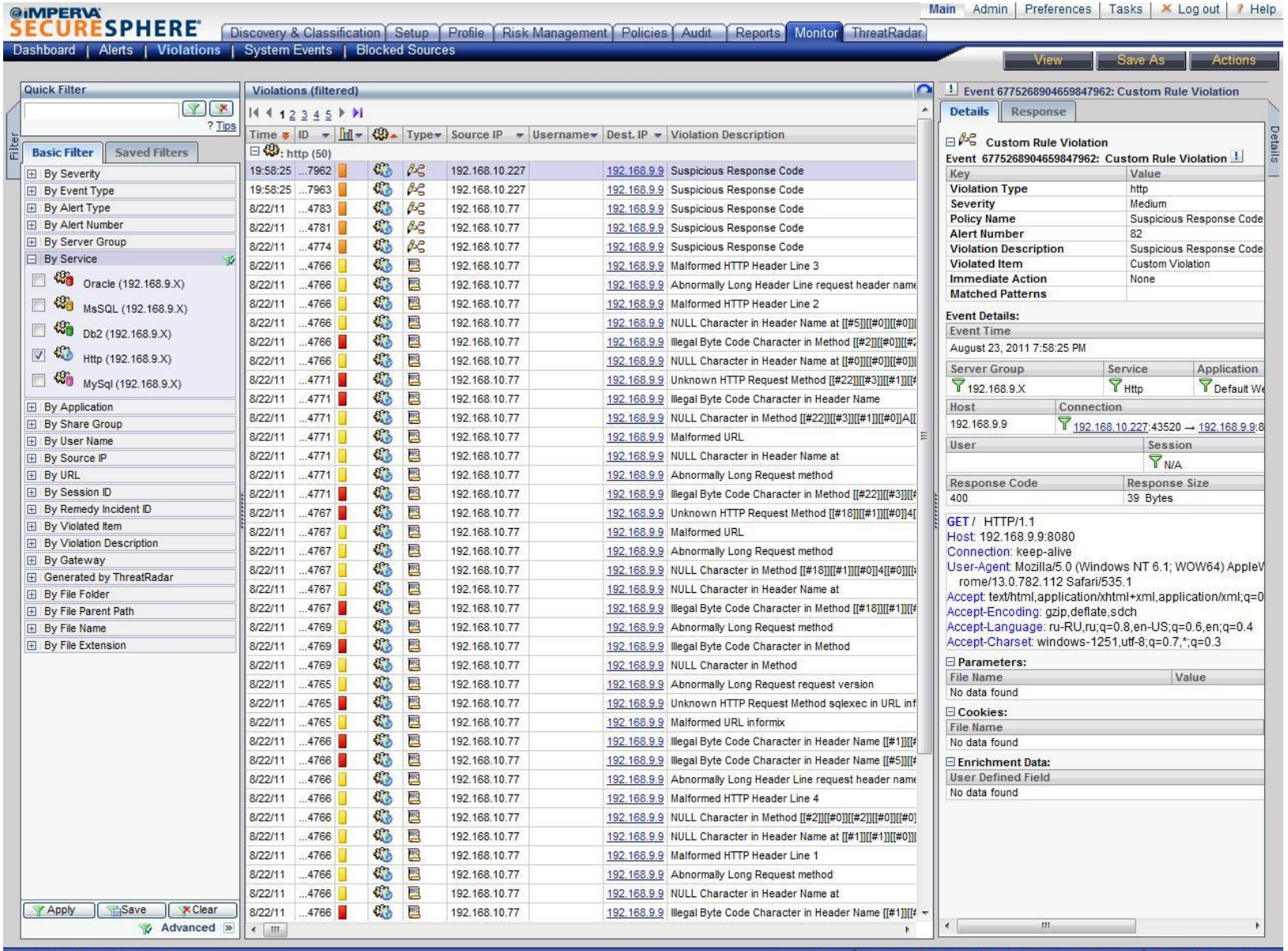
RRC

Тесты СУБД – Монитор/MySQL



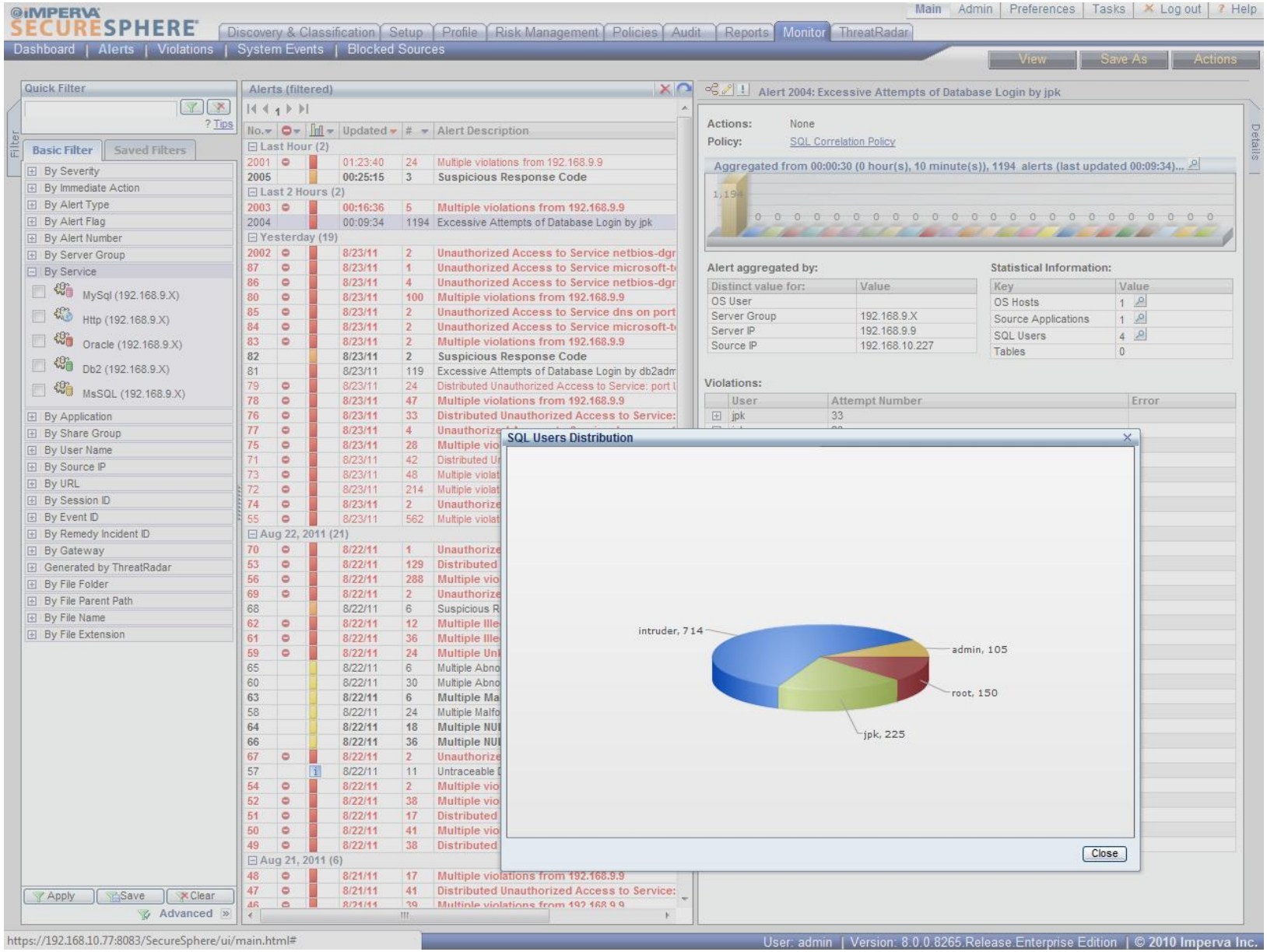
RRC

Тесты СУБД – Монитор/Web



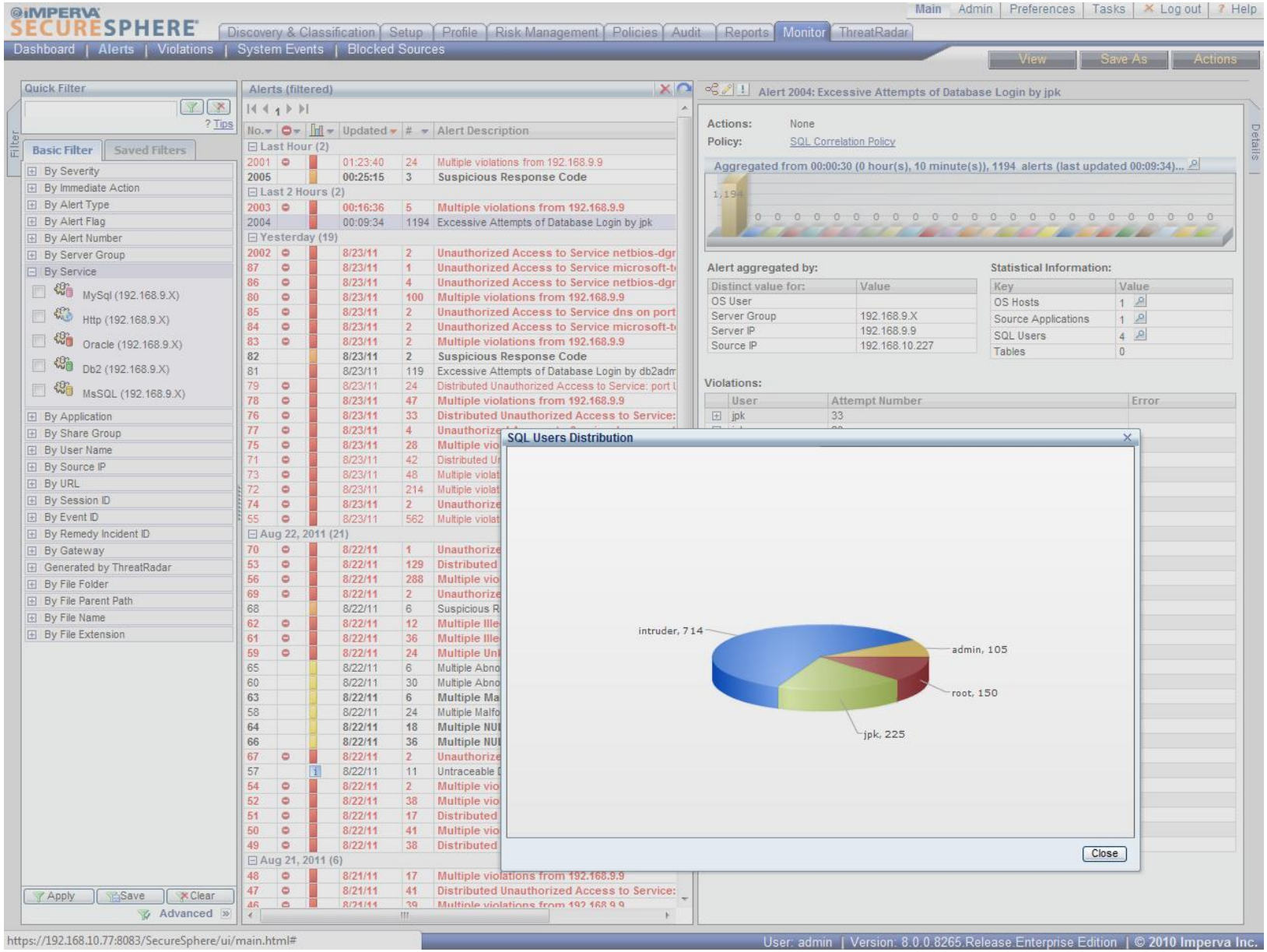
RRC

Тесты СУБД – Монитор/SQL User



RRC

Тесты СУБД – Монитор/SQL User



Тесты связок Web

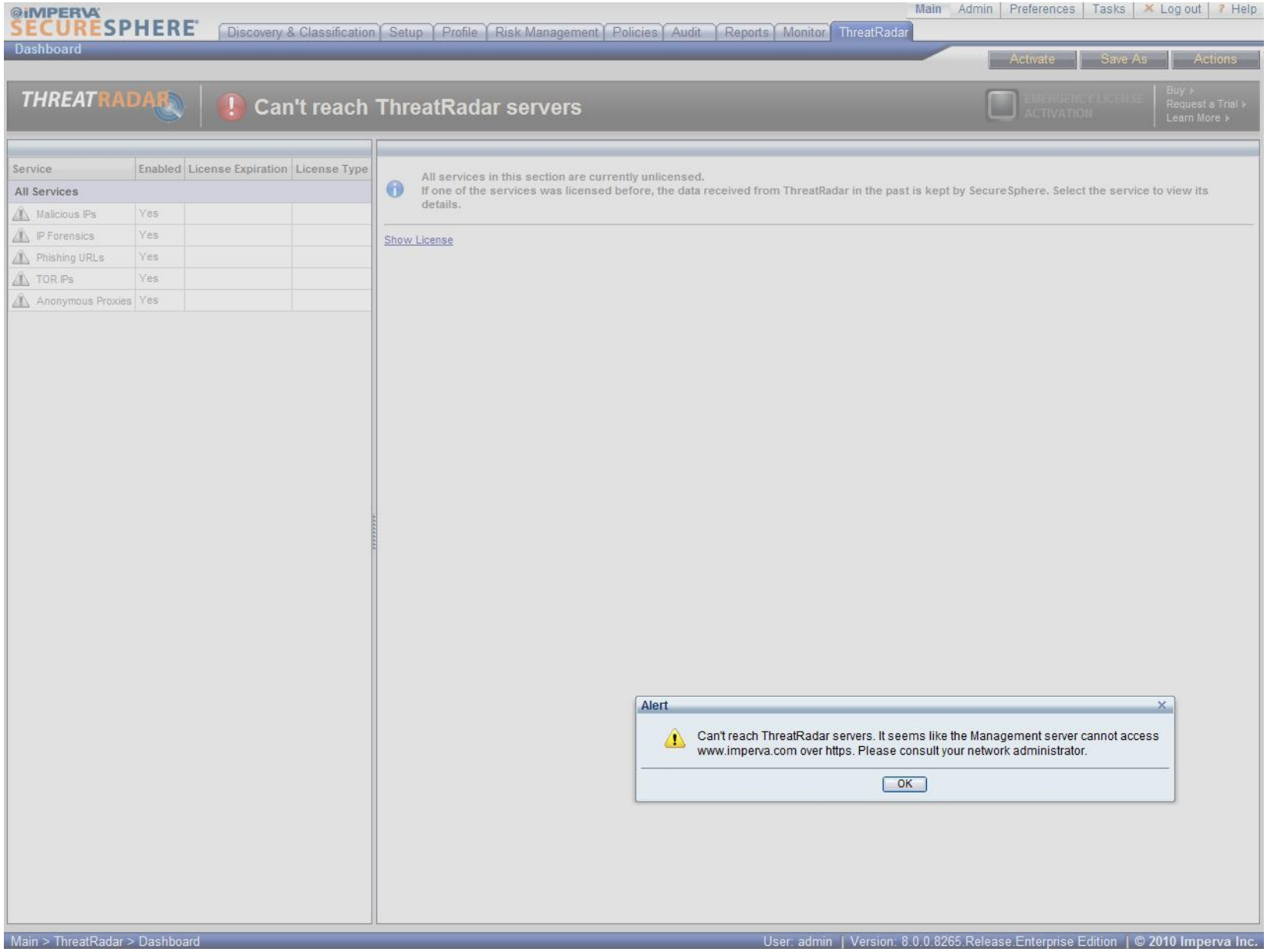
Затем мы плавно перешли к тестированию связок Web приложений, здесь более всего применялся режим автоматизированного сканирования и отслеживалась активность как в мониторе на сервере управления Imperva так и на целевых хостах, посредством мониторинга логов тестируемых служб

Web сервера даже не заметили необычной активности, кроме увеличения количества hits, а пользователи ftp пытающиеся выполнить запрещённые команды после внесения пары настроек более до сервера достучаться не могли, всякие «игры» с SELECT в контексте PHP никаких результатов не принесли

Так же при внедрении не следует забывать о оптимальной, на наш взгляд службе – радар угроз, который позволяет в реальном времени получать информацию и своевременно блокировать трафик «чёрных» ip адресов глобальной сети, узлов TOP, анонимных прокси-серверов, узлов замеченных в фишинге и ip адресов с подозрительной активностью

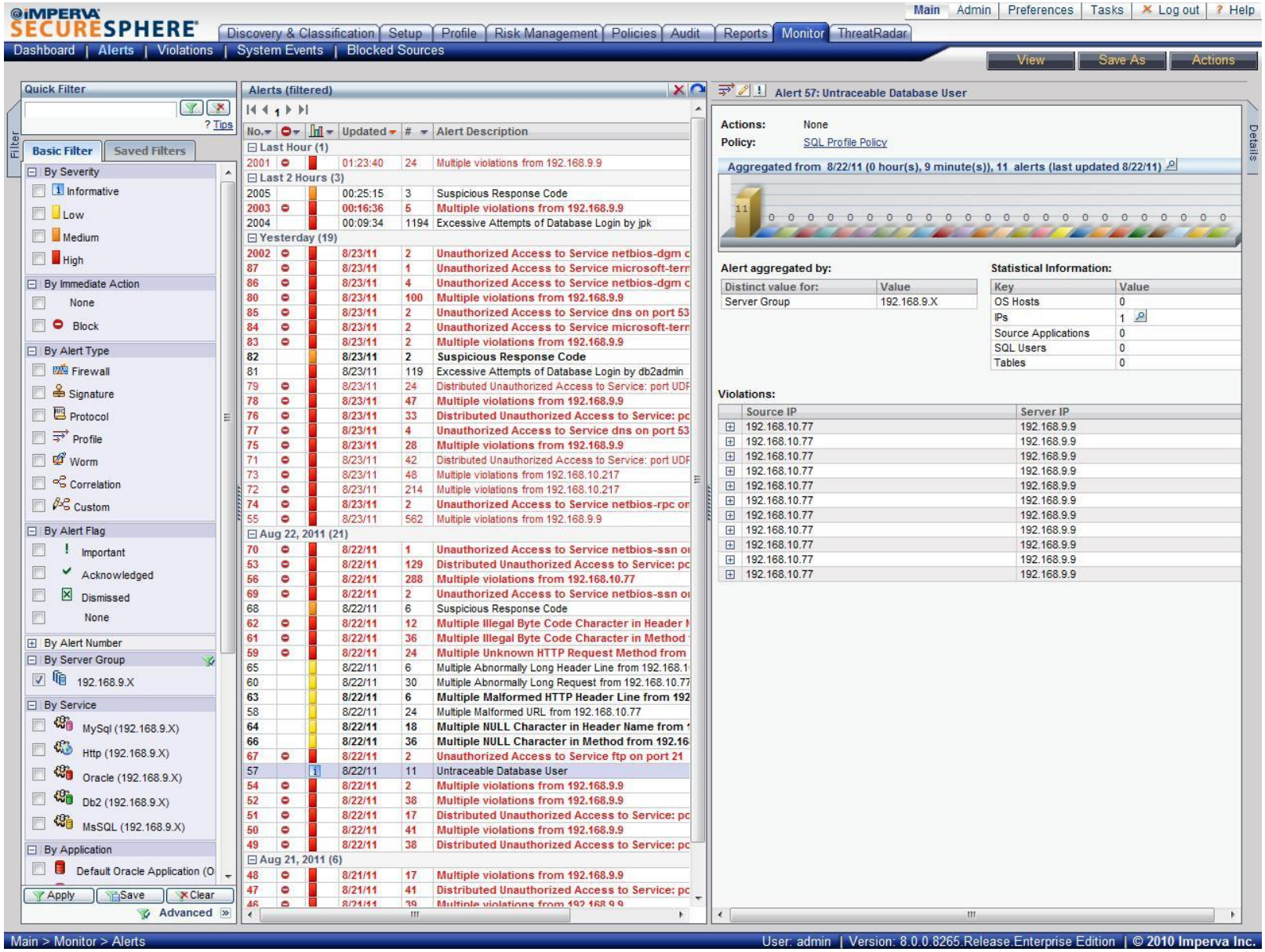
RRC

Радар угроз



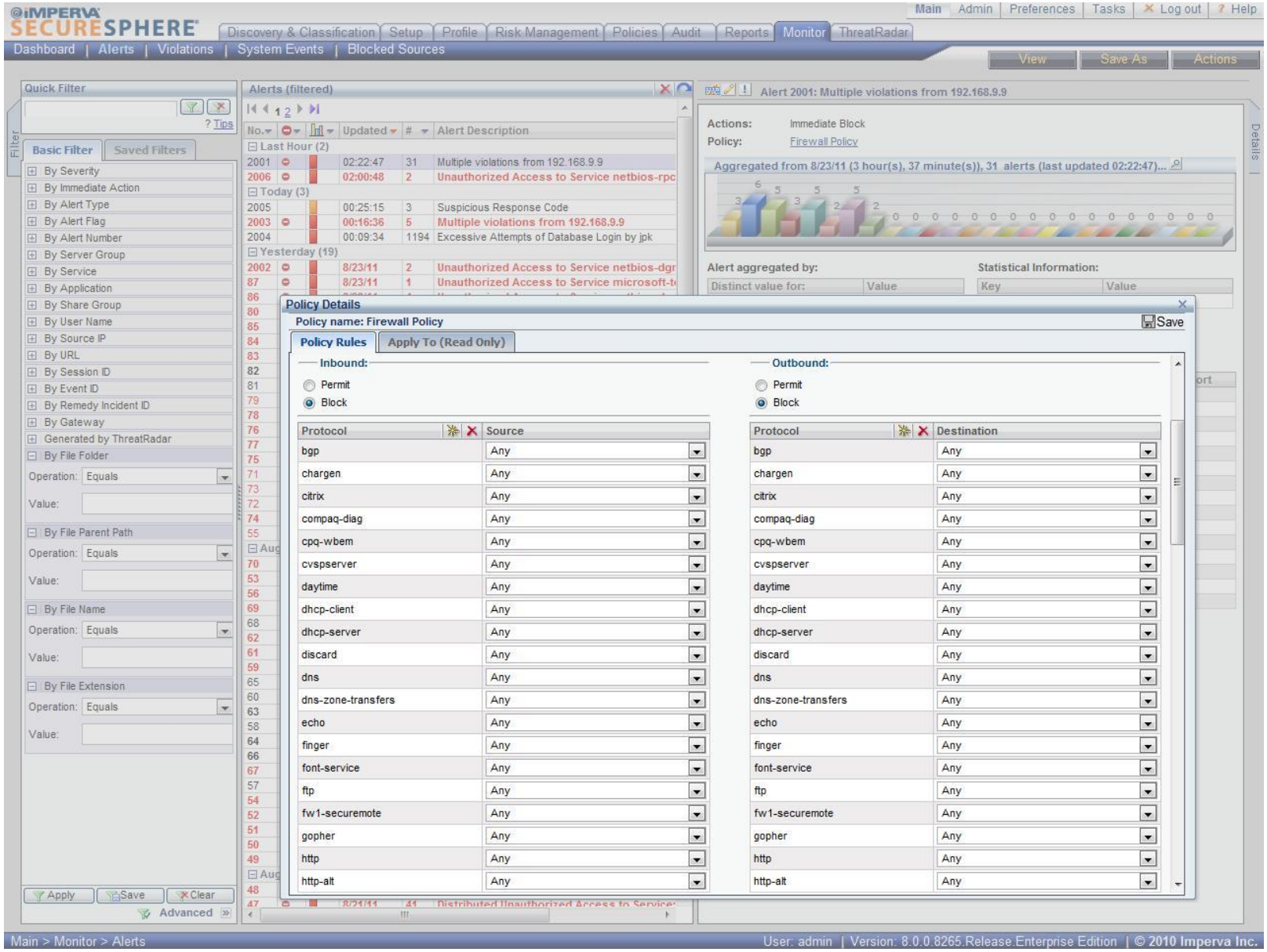
RRC

Монитор – группа серверов



RRC

Монитор – Политики

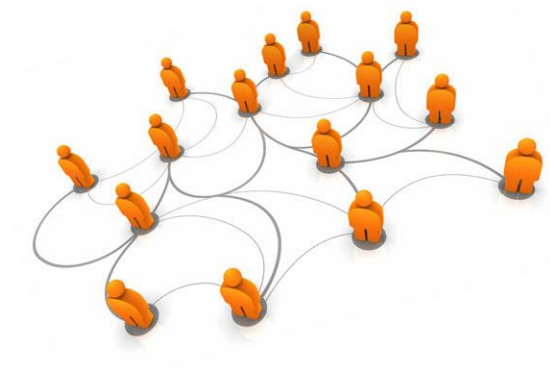


Аудит Инфраструктуры

Чуточку забегая вперёд хотим обратить ваше внимание на возможность выполнения аудита не только для собственных специалистов, но и для аттестации по требуемому уровню, например – выполнив полстью ваш аудит в системе вы сразу можете сопоставить насколько он соответствует требованиям того или иного стандарта, список стандартов на следующем слайде в виде скриншота интерфейса тестируемой системы

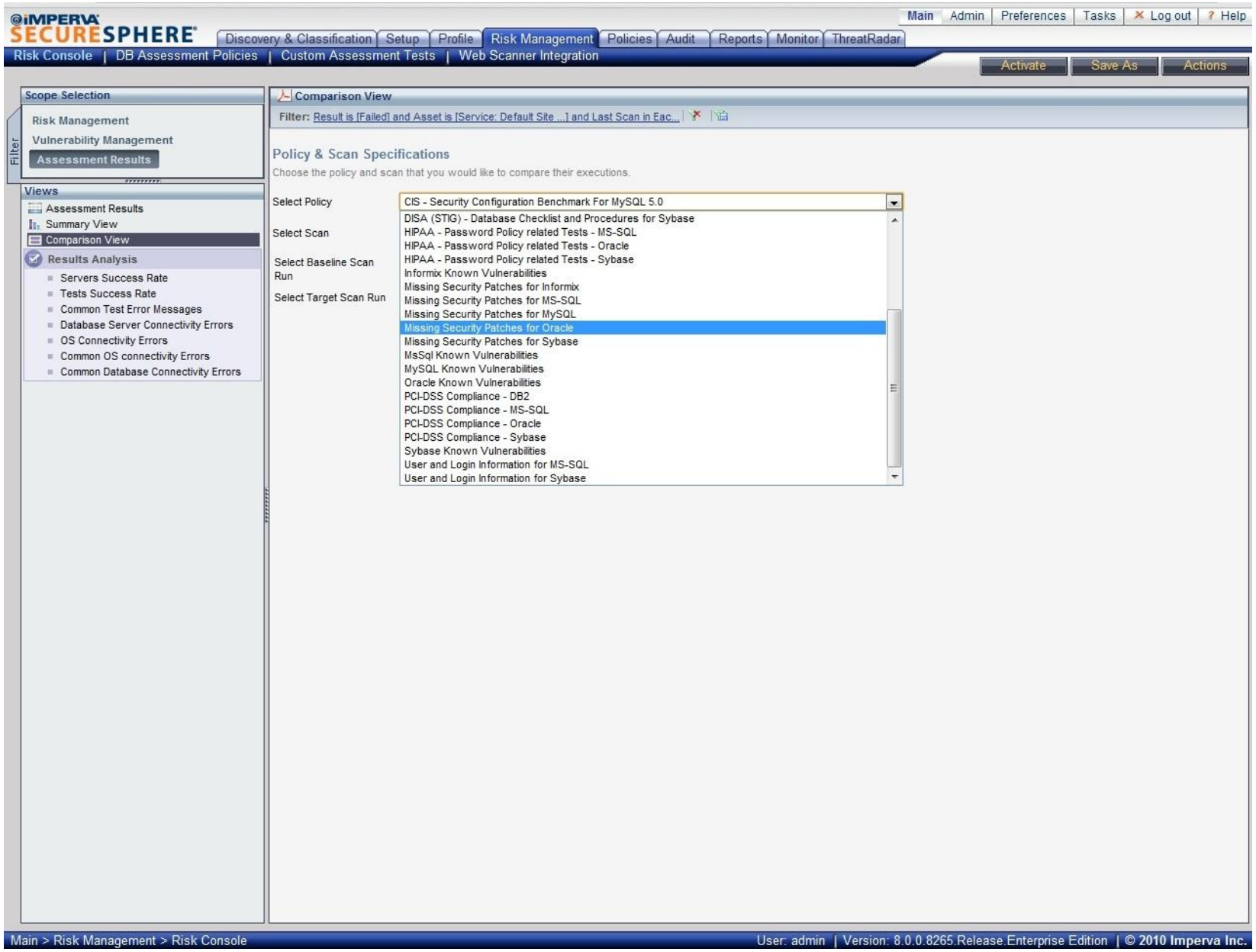
Далее мы рассмотрим возможности системы по пунктам, связывая их с тестами и заостряя внимание на ключевых моментах

Учитывая тот факт, что политики, мы применяли, фактически были политикам по умолчанию, то потенциал системы на самом деле намного больше чем на удалось показать в данной презентации



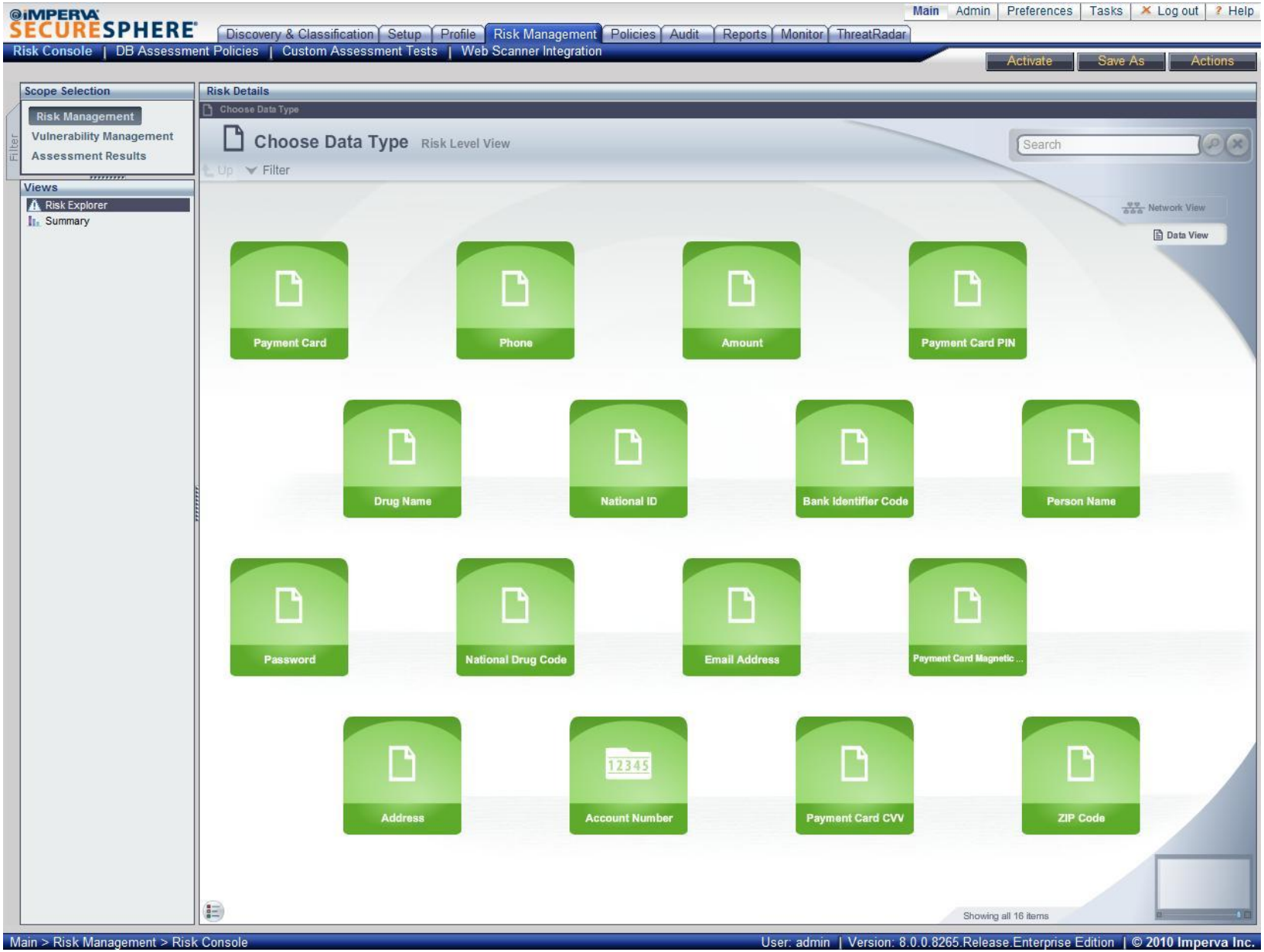
RRC

Аудит – Соответствие стандартам



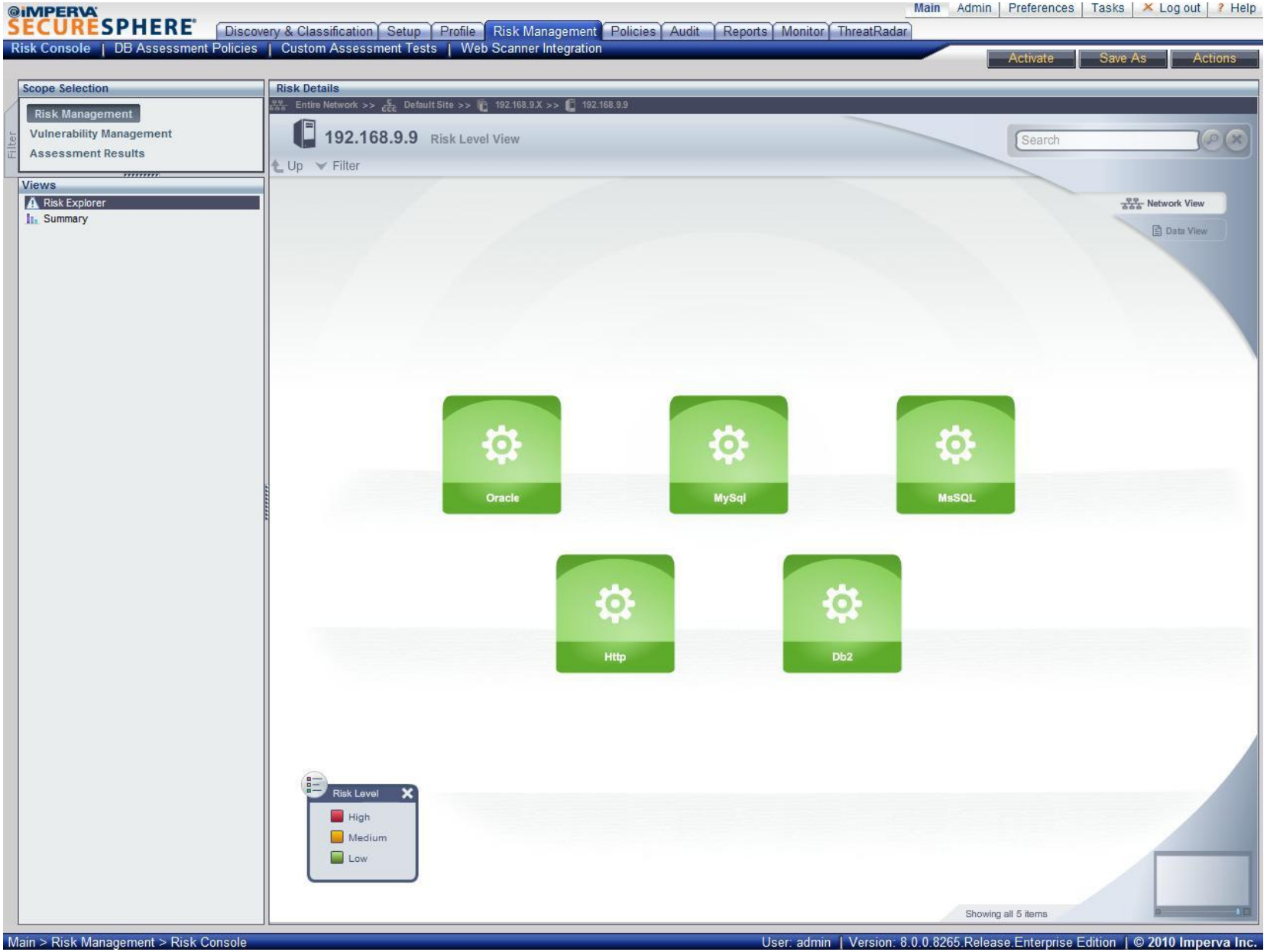
RRC

Управление рисками - Профили



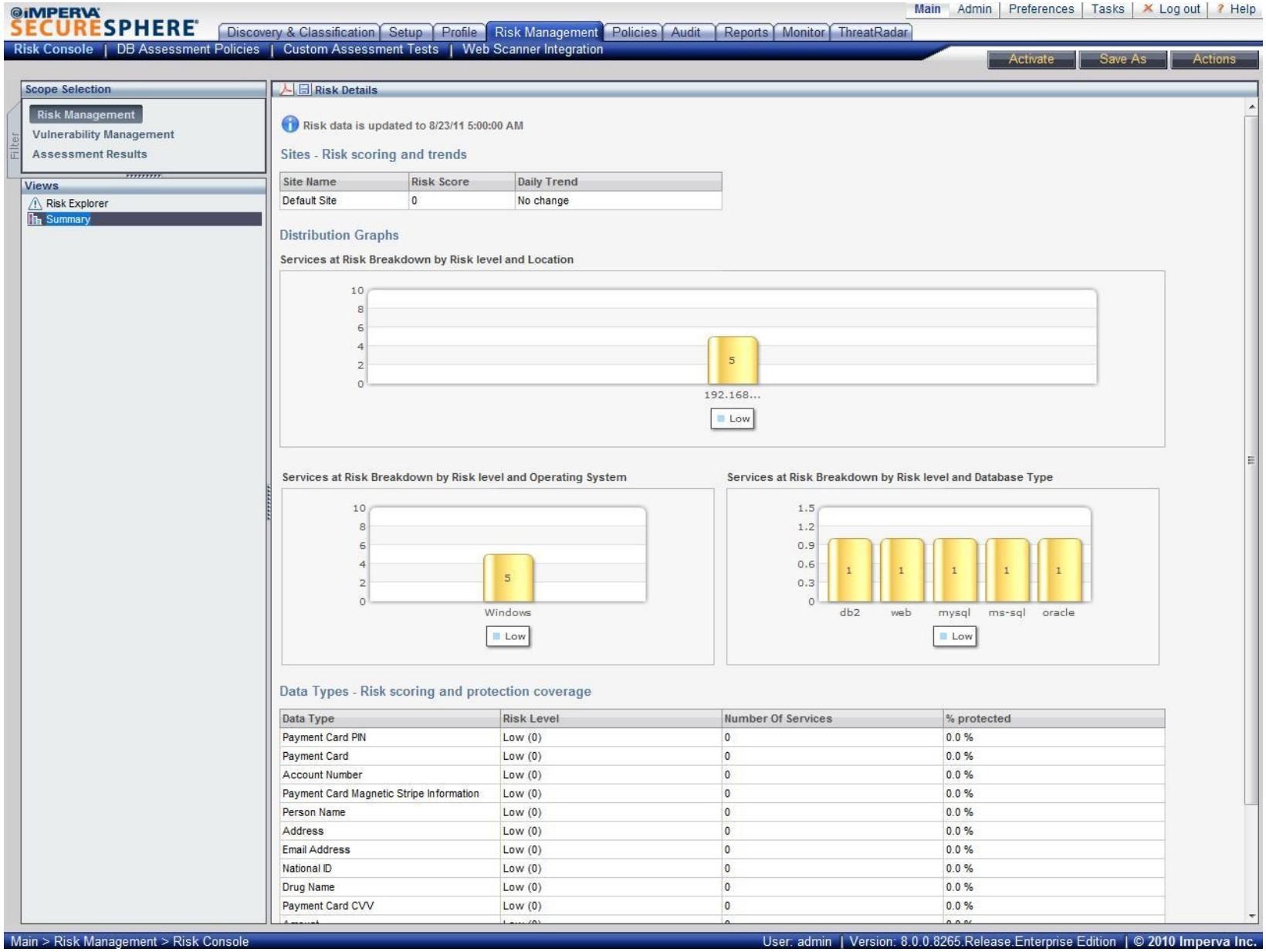
RRC

Управление рисками - Группы



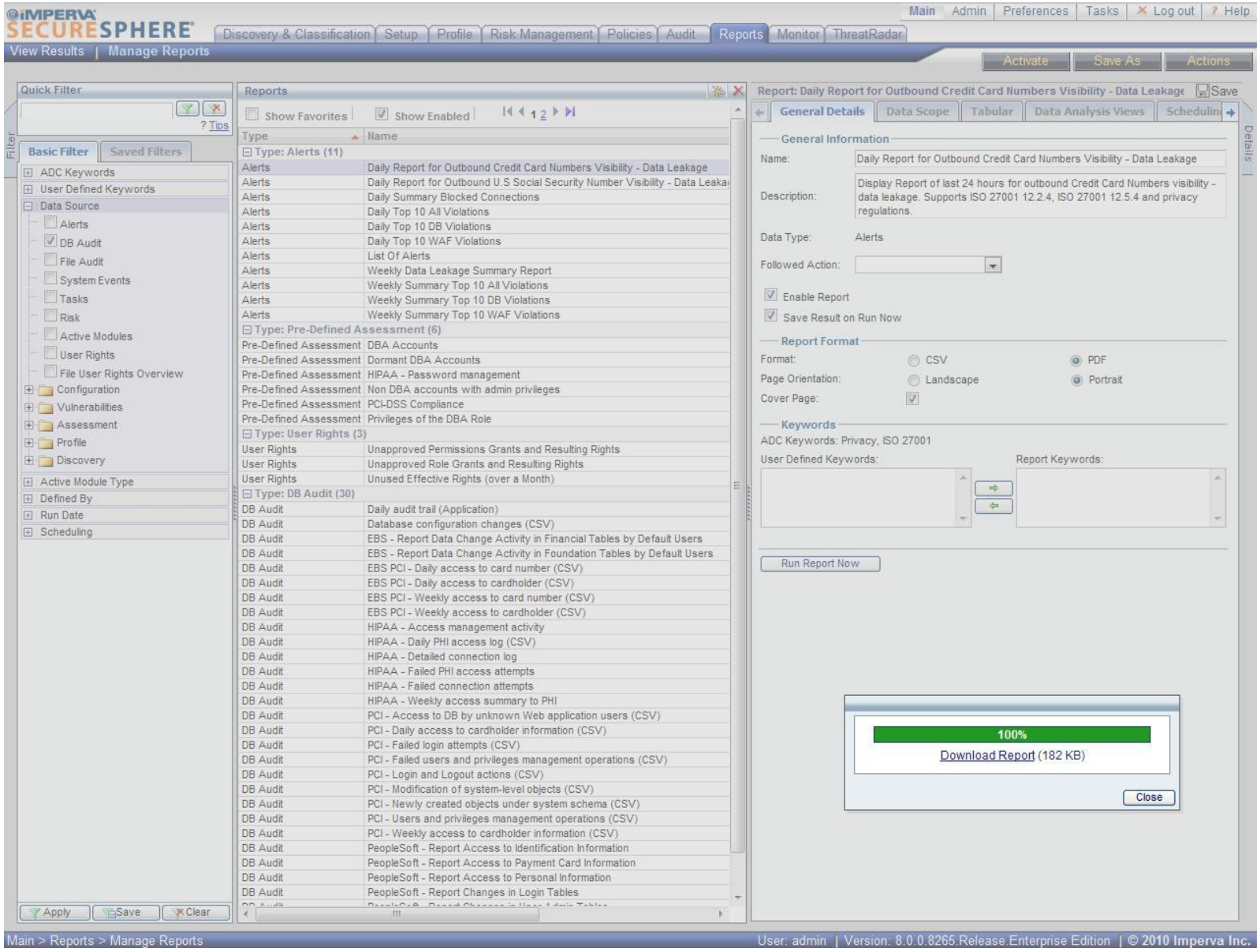
RRC

Управление рисками – В целом



RRC

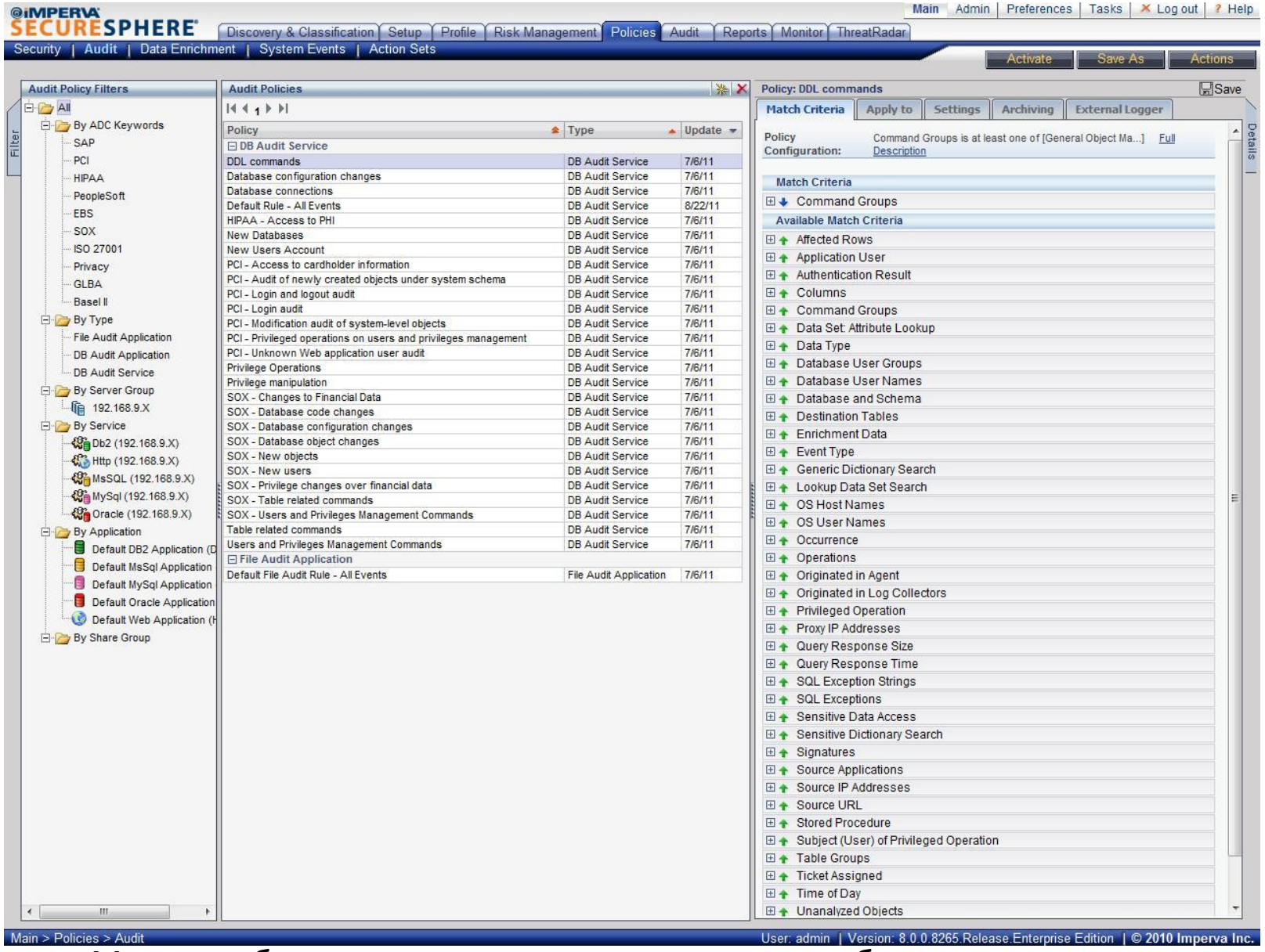
Раздел отчётов



Сгенерированный отчёт доступен в формате pdf

RRC

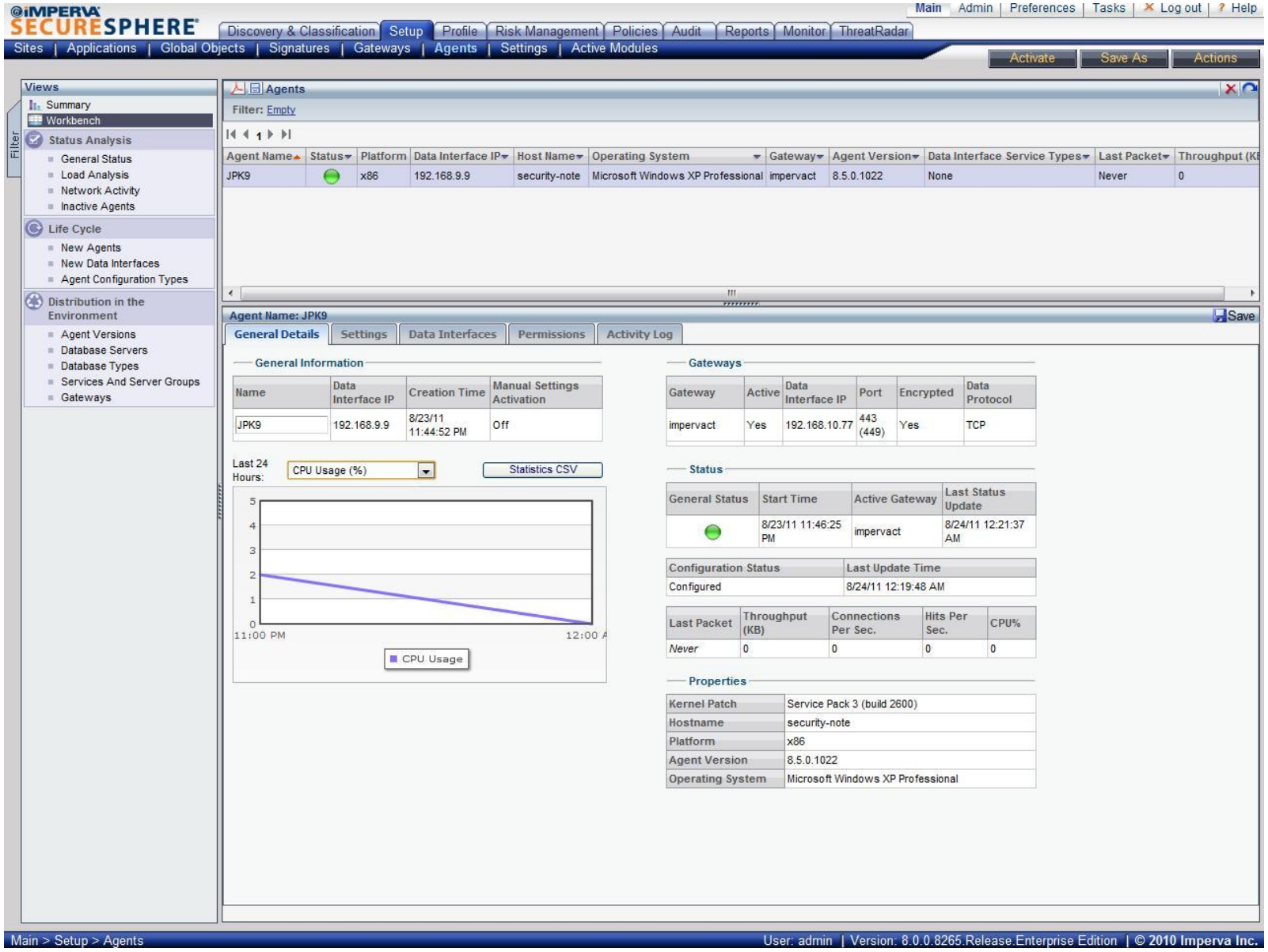
Раздел политик



Можно собрать политику соответствующую требуемому стандарту

RRC

Данные от агентов



Можно выбрать интересующую вас информацию от агента

RRC

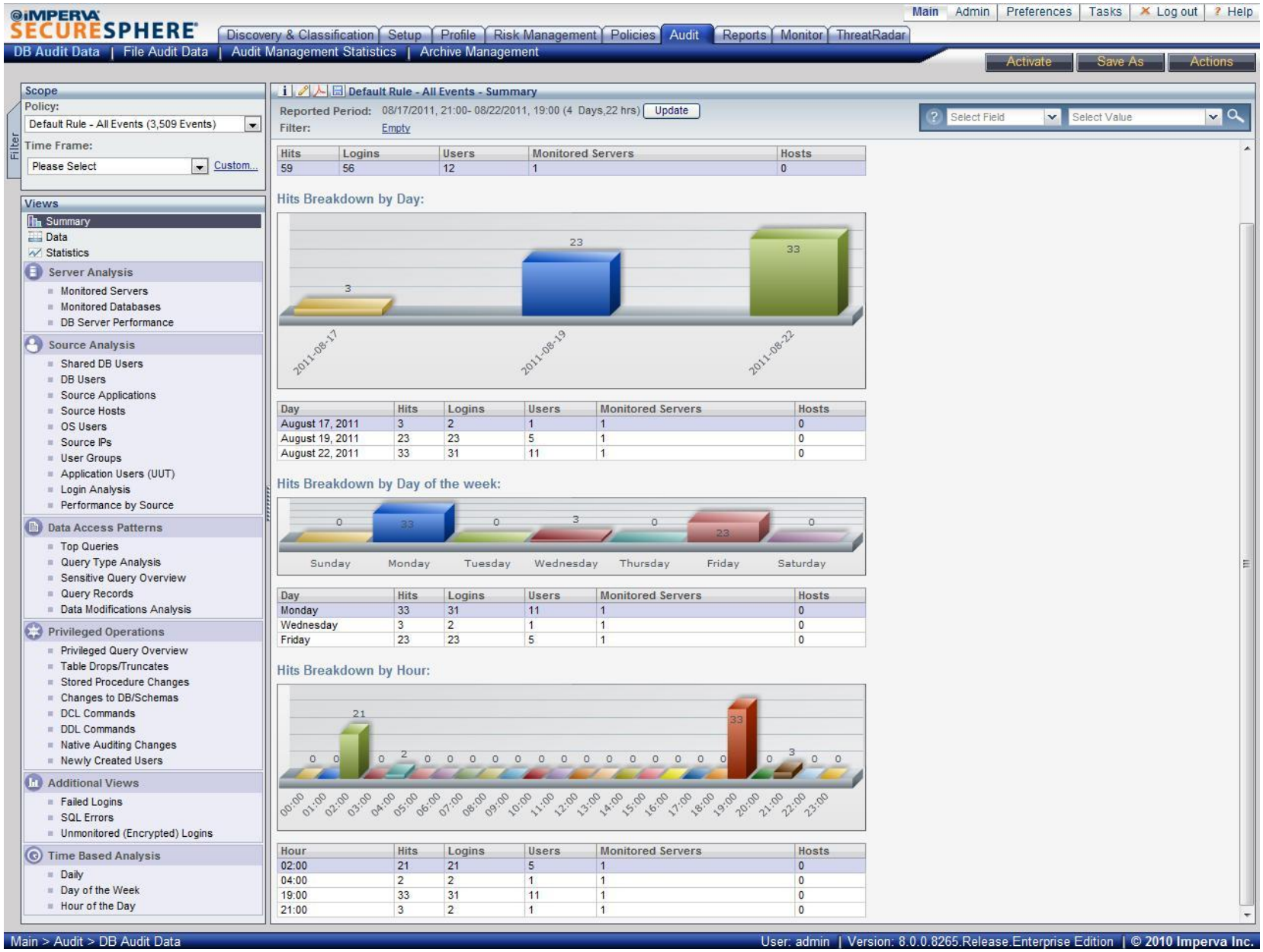
Логи шлюза

```
System log: created: 23/08/2011 15:46:22

Version: 8.5.0.1022

Up Time: 23/08/2011 15:46:22

Log # 1

23/08/2011 15:46:22[info]ImpervaCli.cpp:82 Starting ImpervaCli init
23/08/2011 15:46:22[info]ImpervaCli.cpp:212 Starting target directory init
23/08/2011 15:46:22[info]ImpervaCli.cpp:286 Starting collection table init
23/08/2011 15:46:22[info]ImpervaCli.cpp:324 Collection table init ended successfuly
23/08/2011 15:46:22[info]ImpervaCli.cpp:337 Starting types table init
23/08/2011 15:46:22[info]ImpervaCli.cpp:376 Types table init ended successfuly
23/08/2011 15:46:22[info]ImpervaCli.cpp:389 Starting types table init
23/08/2011 15:46:22[info]ImpervaCli.cpp:150 ImpervaCli init ended successfully
23/08/2011 15:46:26[info]ragentCli.cpp:639 backup of bootstrap.xml was saved in bootstrap.xml.backup22
23/08/2011 15:46:47[NOTIFICATION]PcapProxy.cpp:63 Using WinPcap version 4.1.2
23/08/2011 15:46:50[debug]TcpConnection.cpp:161 Attempt to connect to 192.168.10.77:443
23/08/2011 15:46:50[debug]TcpConnection.cpp:252 Client connected to socket: 188
23/08/2011 15:46:51[debug]SslConnection.cpp:736 SSL socket connect: 188 (192.168.9.9:1384->192.168.10.77:443)
23/08/2011 15:46:51[debug]SslConnection.cpp:424 SSL socket disconnect socket188 (192.168.9.9:1384->192.168.10.77:443)
23/08/2011 15:46:51[debug]TcpConnection.cpp:105 closing socket: 188 (192.168.9.9:1384->192.168.10.77:443)
23/08/2011 15:46:51[debug]TcpConnection.cpp:161 Attempt to connect to 192.168.10.77:443
23/08/2011 15:46:51[debug]TcpConnection.cpp:252 Client connected to socket: 188
23/08/2011 15:46:51[debug]SslConnection.cpp:736 SSL socket connect: 188 (192.168.9.9:1385->192.168.10.77:443)
23/08/2011 15:46:51[debug]SslConnection.cpp:424 SSL socket disconnect socket188 (192.168.9.9:1385->192.168.10.77:443)
23/08/2011 15:46:51[debug]TcpConnection.cpp:105 closing socket: 188 (192.168.9.9:1385->192.168.10.77:443)
23/08/2011 15:46:51[ERROR]RagentReg.cpp:212 Registration failed - bad username or password.
23/08/2011 15:46:51[debug]SslConnection.cpp:420 not connected socket disconnect socket 4294967295 (192.168.9.9:1385->192.168.10.77:443)
23/08/2011 15:47:07[info]ragentCli.cpp:639 backup of bootstrap.xml was saved in bootstrap.xml.backup23
23/08/2011 15:47:33[debug]TcpConnection.cpp:161 Attempt to connect to 192.168.10.77:443
23/08/2011 15:47:33[debug]TcpConnection.cpp:252 Client connected to socket: 188
23/08/2011 15:47:33[debug]SslConnection.cpp:736 SSL socket connect: 188 (192.168.9.9:1386->192.168.10.77:443)
23/08/2011 15:47:33[debug]SslConnection.cpp:424 SSL socket disconnect socket188 (192.168.9.9:1386->192.168.10.77:443)
23/08/2011 15:47:33[debug]TcpConnection.cpp:105 closing socket: 188 (192.168.9.9:1386->192.168.10.77:443)
23/08/2011 15:47:33[debug]TcpConnection.cpp:161 Attempt to connect to 192.168.10.77:443
23/08/2011 15:47:33[debug]TcpConnection.cpp:252 Client connected to socket: 188
23/08/2011 15:47:33[debug]SslConnection.cpp:736 SSL socket connect: 188 (192.168.9.9:1387->192.168.10.77:443)
23/08/2011 15:47:33[debug]SslConnection.cpp:424 SSL socket disconnect socket188 (192.168.9.9:1387->192.168.10.77:443)
23/08/2011 15:47:33[debug]TcpConnection.cpp:105 closing socket: 188 (192.168.9.9:1387->192.168.10.77:443)
23/08/2011 15:47:33[ERROR]RagentReg.cpp:212 Registration failed - bad username or password.
23/08/2011 15:47:33[debug]SslConnection.cpp:420 not connected socket disconnect socket 4294967295 (192.168.9.9:1387->192.168.10.77:443)
23/08/2011 15:47:52[info]ragentCli.cpp:639 backup of bootstrap.xml was saved in bootstrap.xml.backup24
23/08/2011 15:48:19[debug]TcpConnection.cpp:161 Attempt to connect to 192.168.10.77:443
23/08/2011 15:48:19[debug]TcpConnection.cpp:252 Client connected to socket: 188
23/08/2011 15:48:19[debug]SslConnection.cpp:736 SSL socket connect: 188 (192.168.9.9:1388->192.168.10.77:443)
23/08/2011 15:48:19[debug]SslConnection.cpp:424 SSL socket disconnect socket188 (192.168.9.9:1388->192.168.10.77:443)
23/08/2011 15:48:19[debug]TcpConnection.cpp:105 closing socket: 188 (192.168.9.9:1388->192.168.10.77:443)
```

Можно загрузить через веб-интерфейс

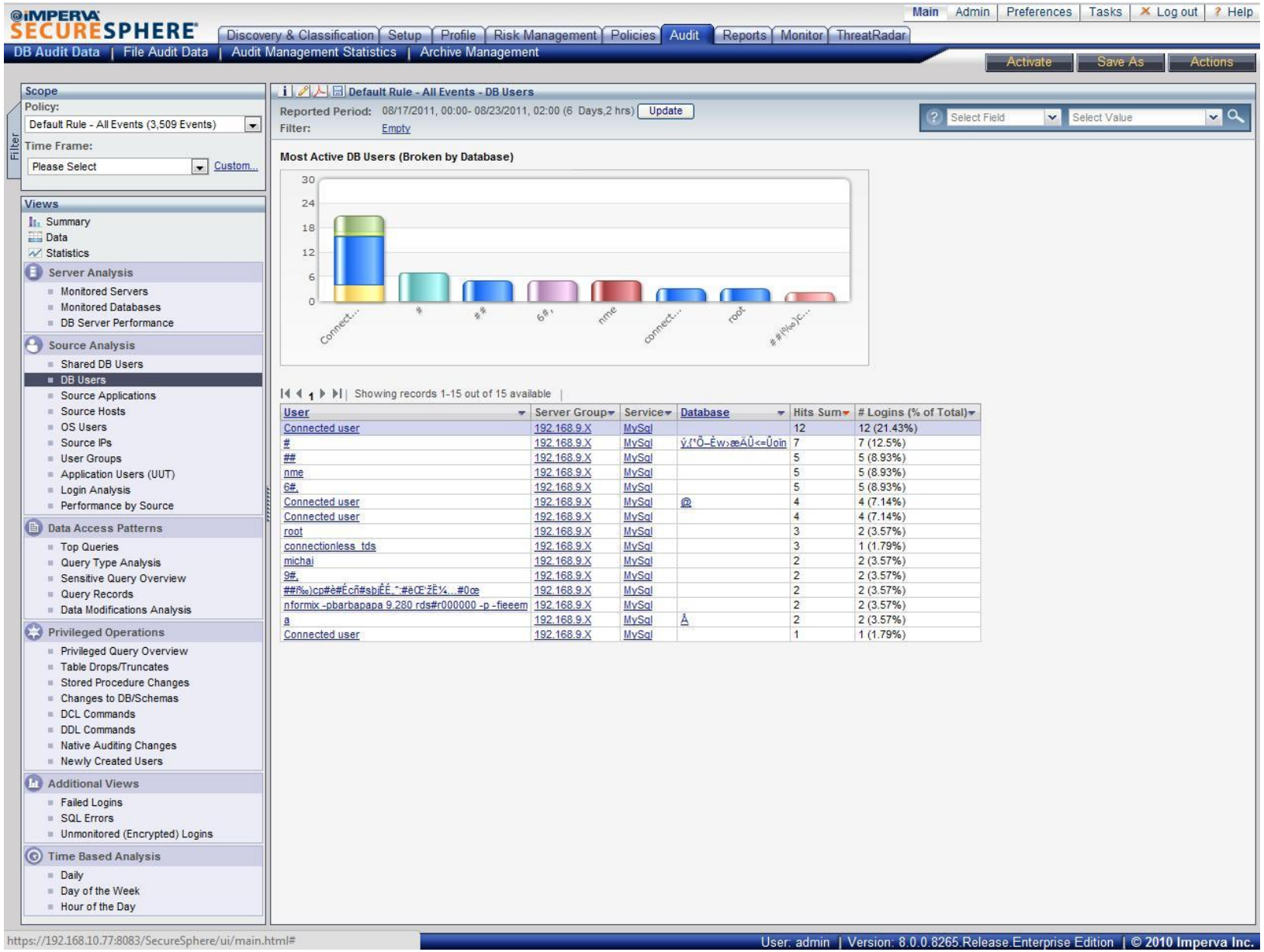
RRC

Аудит – в целом



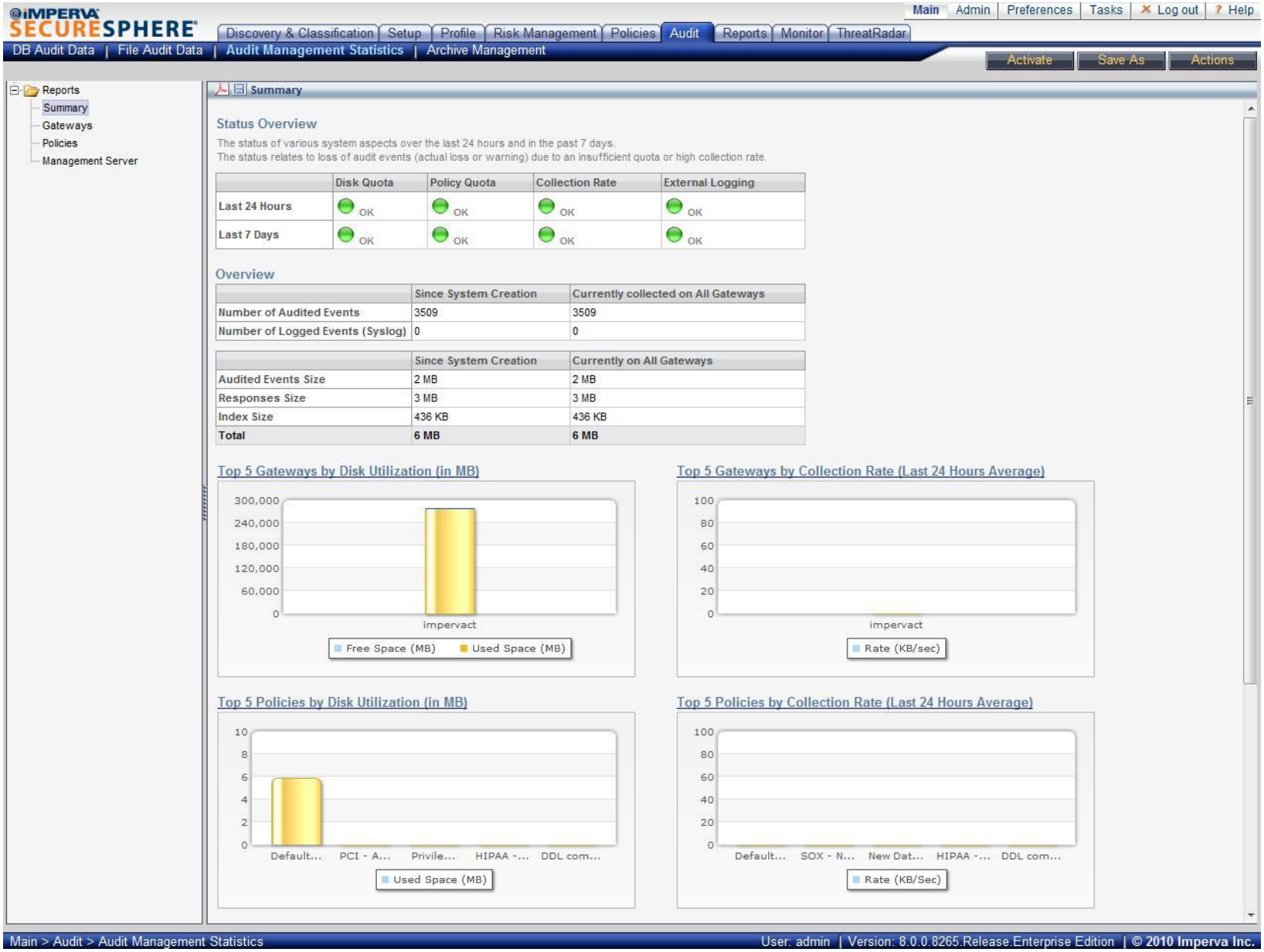
RRC

Аудит – пользователи



RRC

Аудит – по шлюзам

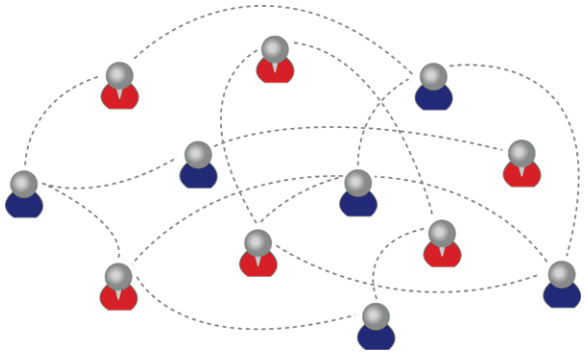


RRC

Заключение

Основные аргументы в пользу данной системы:

- способна сразу после установки и настройки сетевых параметров с политиками по умолчанию «приступить» к защите вашего внутреннего периметра
- шлюз можно устанавливать в режиме прозрачного моста, что никак не повлияет на текущую конфигурацию вашей сети
- благодаря постоянным автоматическим обновлениям сигнатур и комплексу «радар угроз» вам не нужно будет беспокоится о обновлении ваших СУБД и веб-сервисов исключительно из соображений безопасности
- благодаря встроенным средствам и политикам вы можете поддерживать свою систему в соответствии с требуемым стандартом безопасности
- вы получаете великолепную систему мониторинга и оповещения, у ваших специалистов всегда будет актуальная информация



Ваши вопросы

Благодарим за внимание!