КОРПОРАТИВНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ: ТЕСТИРОВАНИЕ НА ПРОНИКНОВЕНИЕ

СЦЕНАРИИ АТАК

2017

POSITIVE TECHNOLOGIES

Содержание

Введение

Ежегодно в корпоративных информационных системах (КИС) различных компаний обнаруживается множество опасных уязвимостей, которые позволяют внешнему нарушителю получать доступ к критически важным бизнес-системам в локальной вычислительной сети (ЛВС), а внутренним злоумышленникам — развивать атаку до получения полного контроля над всей КИС[1]. В случае успеха подобные атаки приводят к существенным финансовым и репутационным потерям[2].

В целях предотвращения таких угроз эксперты Positive Technologies ежегодно проводят десятки тестирований на проникновение для крупнейших компаний как в России, так и за рубежом. При тестировании моделируется поведение потенциальных нарушителей, что позволяет оценить реальный уровень безопасности системы и выявить конкретные недостатки механизмов защиты, в том числе и те, которые могут остаться незамеченными при использовании других методов аудита.

В данном отчете представлены типовые сценарии атак, которые успешно моделировались в наших тестированиях на проникновение за последние три года. Описанные сценарии позволяют получать 100%-й контроль над ЛВС в рамках всех тестирований от лица внутреннего нарушителя. Что касается тестирований от лица внешнего нарушителя, преодолеть периметр удается примерно в 80% проектов. В отчете не раскрываются адреса ресурсов, имена сотрудников, контактная информация и прочие данные, которые позволили бы определить протестированные организации. Кроме того, описанные сценарии атак не привязаны к сфере деятельности компаний, так как используемые для атак недостатки защиты могут быть отнесены к организациям любой отрасли.

Аналогичные техники применяют и реальные злоумышленники для целевых атак. В частности, результаты расследований компьютерных инцидентов, проведенных нашими экспертами в 2016 году, свидетельствуют о том, что киберпреступники стали реже использовать сложные атаки с эксплуатацией ранее неизвестных уязвимостей (0-day) — вместо этого они чаще применяют более простые методы, для которых не требуются значительные финансовые затраты. Более того, злоумышленники все чаще используют общедоступные и коммерческие инструменты (например, ПО для проведения легальных тестов на проникновение — Cobalt Strike, Metasploit и т. п.), а также встроенные функции ОС, что позволяет им скрывать свою активность в инфраструктуре. Яркими примерами могут служить атаки на банки России и СНГ, осуществленные хакерской группой Cobalt[3], а также группами Carbanak[4] и Buhtrap[5].

Важно понимать, что уязвимости, эксплуатируемые в таких атаках, характерны не только для банков — они могут присутствовать в КИС любой организации. Методы тестирования на проникновение, описанные в данном отчете, позволяют выявить эти уязвимости до того, как ими воспользуются злоумышленники. После каждого сценария атаки в отчете даются рекомендации по необходимым мерам защиты, благодаря которым администраторы КИС и специалисты по информационной безопасности могут существенно повысить общий уровень защищенности корпоративной инфраструктуры от атак со стороны внешнего и внутреннего нарушителя.

[1] https://www.ptsecurity.com/upload/ptru/analytics/Corporate-Vulnerability-2015-rus.pdf
[2] https://www.ptsecurity.com/upload/corporate/ru-ru/analytics/Cybersecurity-2016-2017-rus.pdf
[3] https://www.ptsecurity.com/upload/ptru/analytics/Cobalt-Snatch-rus.pdf
[4] http://www.group-ib.com/files/Anunak_APT_against_financial_institutions.pdf
[5] http://www.group-ib.ru/brochures/gib-buhtrap-report.pdf

Преодоление периметра КИС

На основе многолетнего опыта тестов на проникновение мы выделяем шесть основных техник атак, которые могут быть использованы внешним нарушителем для преодоления сетевого периметра КИС и получения доступа к ЛВС. Эти техники основаны на эксплуатации следующих типов уязвимостей, характерных для КИС любой организации:

+ недостатки управления учетными записями и паролями;
+ уязвимости веб-приложений;
+ недостатки фильтрации трафика;
+ недостатки управления уязвимостями и обновлениями;
+ плохая осведомленность пользователей в вопросах информационной безопасности;
+ недостатки конфигурации и разграничения доступа.

Для преодоления периметра сети нарушителю необходимо иметь возможность выполнять команды операционной системы (ОС) на атакованном узле. Перечисленные сценарии атак позволяют не только получить такие привилегии, но и полностью скомпрометировать сервер. При наличии на уязвимом сервере интерфейса внутренней сети, нарушитель сможет развивать атаки на ресурсы ЛВС.

Важно отметить, что в отдельных проектах по тестированию на проникновение каждый из перечисленных сценариев позволял достичь поставленной цели без осуществления других атак. Иногда для преодоления периметра необходимо было использовать комбинацию представленных методов, что лишь повышало сложность атаки, но не влияло на ее успешность.

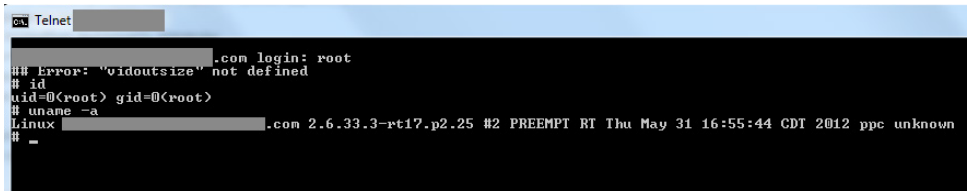
Сценарий 1. Подбор учетных данных

Интерфейсы управления и удаленного доступа

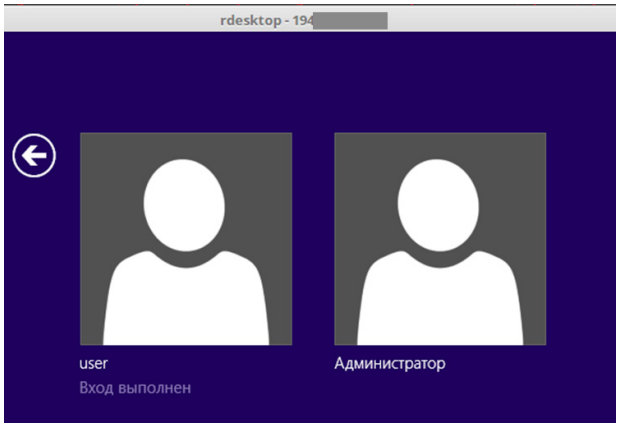
Не секрет, что администрировать сложные распределенные системы только с помощью локальных подключений — слишком трудоемкая задача. Современные технологии позволяют упростить работу системного администратора с помощью протоколов удаленного управления устройствами, таких как Telnet, RSH, SSH, а также протоколов для удаленного подключения, таких как RDP. Часто администраторы используют общедоступное ПО для удаленного подключения — RAdmin, Ammyy Admin и т. п. Использование таких инструментов позволяет внешнему нарушителю проводить атаки на подбор учетных данных и в случае успеха получать доступ к ОС этих устройств.

Для реализации атаки злоумышленнику не нужно обладать специальными знаниями и навыками. В большинстве случаев достаточно ноутбука, программы для подбора учетных данных (легко найти в открытом доступе, например Hydra) и словаря (в Интернете можно найти множество готовых словарей идентификаторов и паролей пользователей, наиболее распространенных для каждой конкретной системы или службы). В случае фильтрации подключений по IP-адресам атака будет затруднена, однако нарушитель, скорее всего, найдет другие пути, например, скомпрометирует другие узлы на периметре и попробует развить атаку не со своего адреса, а с этих скомпрометированных узлов, либо применит другие методы обхода фильтрации.

Примерами распространенных комбинаций идентификатора и пароля для доступа по SSH и Telnet являются root:root, root:toor, admin:admin; test:test. Иногда можно получить доступ с максимальными привилегиями и вовсе без ввода пароля.



Для доступа по RDP используются локальные либо доменные учетные записи, среди которых часто встречаются Administrator:P@ssw0rd; Administrator:123456; Administrator:Qwerty123, а также гостевая учетная запись Guest с пустым паролем.



Рекомендации по защите. Для повышения безопасности в случае организации удаленного доступа по протоколу SSH рекомендуется использование аутентификации по ключу, когда на сервере хранится открытый ключ клиента, а на стороне клиента — закрытый. Только обладая закрытым ключом, клиент может авторизоваться на сервере. В целом же рекомендуется вовсе ограничивать доступ к узлам по протоколам управления из сети Интернет, разрешая такие подключения только из ЛВС с ограниченного числа рабочих станций администраторов. Для этого необходимо применить соответствующие настройки на межсетевом экране. Кроме того, рекомендуется внедрить строгую парольную политику для исключения возможности установки простых или словарных паролей. Если же необходимо администрировать ресурсы удаленно, рекомендуется использовать защищенное подключение по технологии VPN.

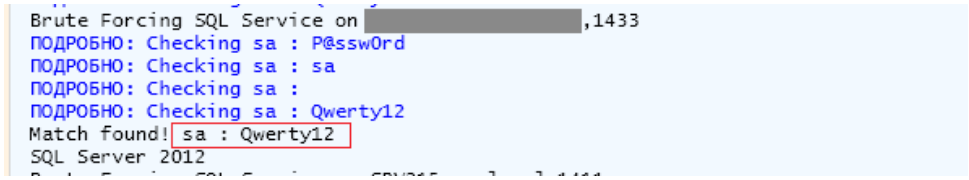
Интерфейсы администрирования веб-серверов и СУБД

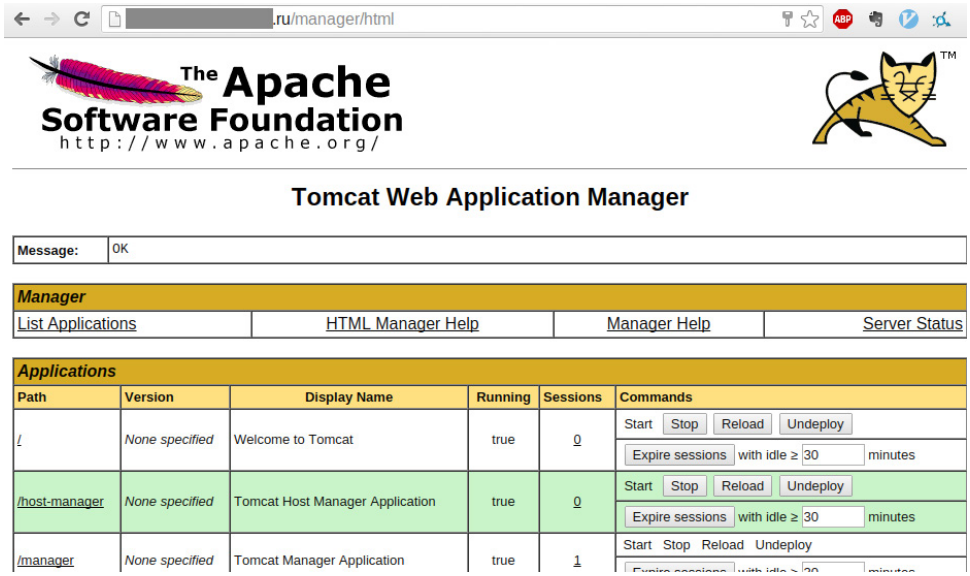
Существуют и другие службы, доступ к которым может позволить внешнему нарушителю получить полный контроль над узлом на периметре и возможность развивать атаки на ресурсы ЛВС. К ним можно отнести СУБД и веб-серверы.

Если для управления серверами по протоколам SSH, Telnet и т. п. необходимо изначально задать определенное значение пароля вручную, то для различных СУБД и веб-серверов, устанавливаемых «из коробки», обычно существуют стандартные (установленные вендором по умолчанию) учетные данные. Конечно, в документации к системам производители настоятельно рекомендуют сменить стандартный пароль при первом же подключении. Однако, как показывает практика, далеко не все администраторы следуют этим указаниям, а многие изменяют учетные данные на столь же простые комбинации.

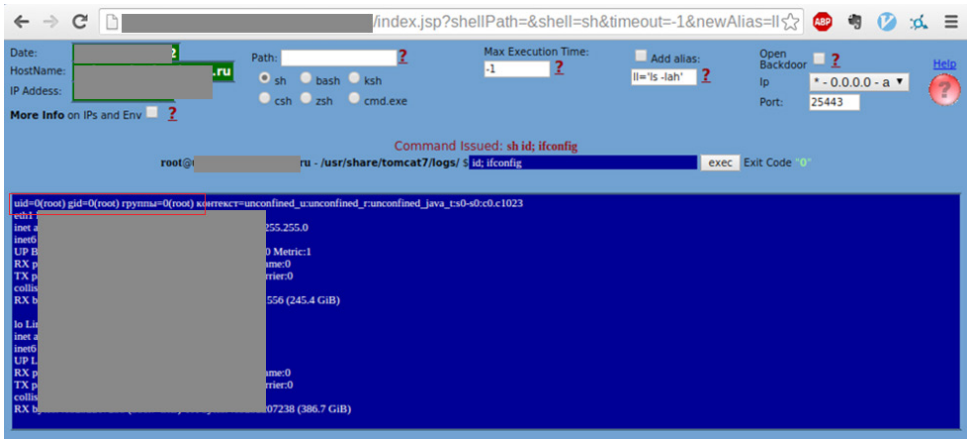
Примеры наиболее распространенных учетных данных, которые выявляются в наших тестах на проникновение:

+ для СУБД — sa:sa, sa:P@ssw0rd; oracle:oracle; postgres:postgres; mysql:mysql, mysql:root; различные комбинации с пустым паролем;
+ для серверов Tomcat — tomcat:tomcat, tomcat:admin.





Интерфейс администрирования Tomcat Web Application Manager позволяет загружать файлы в формате архива с расширением .war. Атакующий может загрузить не только веб-приложение, но и веб-интерпретатор командной строки, и получить возможность выполнять команды ОС.

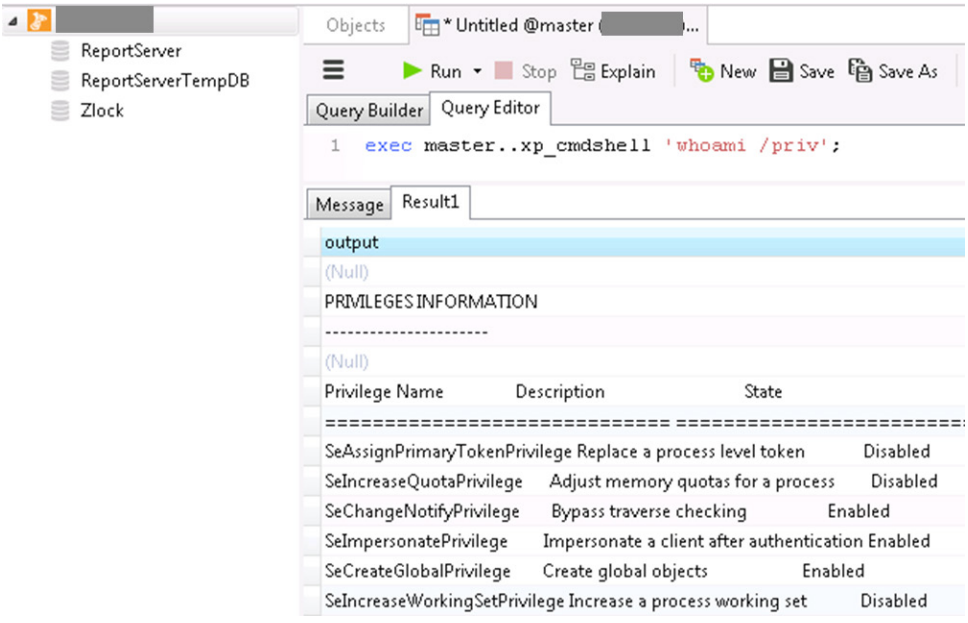


Обладая доступом к СУБД, нарушитель может не только получать информацию из базы данных, но и выполнять команды ОС на сервере с привилегиями СУБД. Другими словами, нарушитель получает доступ к серверу, аналогичный тому, который может быть получен по интерфейсам управления, разница может быть лишь в уровне привилегий. Если эти привилегии ограничены, злоумышленник может попытаться использовать уязвимости

ОС с целью повышения своей роли в системе, однако для развития атак на ресурсы ЛВС текущего уровня может оказаться достаточно.

```
                    $ psql -h [redacted] -U postgres
psql (9.1.14, server 8.4.4)
WARNING: psql version 9.1, server version 8.4.
         Some psql features might not work.
Type "help" for help.

postgres=# \l
                                                List of databases
   Name    |      Owner      | Encoding |        Collate        |         Cty
-----------+-----------------+----------+-----------------------+------------
 [redacted] | postgres        | WIN1251  | Russian_Russia.1251 | Russian_Ru
 postgres  | [redacted]      | ...      | WIN1251  | Russian_Russia.1251
 template0 |                 | ...      | WIN1251  | Russian_Russia.1251
           |                 |          |                       |
 template1 |                 | ...      | WIN1251  | Russian_Russia.1251
           |                 |          |                       |
(4 rows)

postgres=#
```

Уровень привилегий веб-сервера, СУБД и отдельных пользователей зачастую является ключевым вопросом при решении задачи защиты ресурсов на сетевом периметре. К примеру, в старых версиях MS SQL Server продукт устанавливался в ОС по умолчанию с привилегиями NT AUTHORITY\SYSTEM, максимальными в Windows. Нарушитель, подобравший учетную запись СУБД, моментально получал полный контроль над сервером.



В актуальных версиях MS SQL Server этот недостаток был учтен, привилегии СУБД по умолчанию стали ограниченными — NT SERVICE\MSSQLSERVER[6]. Однако даже эти ограничения зачастую не обеспечивают должный уровень защиты. В одном из проектов по тестированию на проникновение мы выяснили, что пользователь NT SERVICE\MSSQLSERVER обладает привилегиями SeImpersonatePrivilege в ОС, которые позволяют ему с помощью

[6] https://msdn.microsoft.com/en-us/library/ms143504.aspx#Serv_Perm

токена делегирования (impersonation-token[7]) присвоить себе привилегии любого другого пользователя из перечня доступных. Для этого может быть использована утилита Mimikatz. В ходе тестирования было установлено, что для присвоения доступны максимальные привилегии в системе NT AUTHORITY\SYSTEM.

Рекомендации по защите. Администраторам необходимо тщательно следить за тем, какой уровень привилегий используют те или иные системы и пользователи, и минимизировать уровень таких привилегий.

Рекомендуется ограничивать доступ к СУБД и интерфейсам администрирования веб-серверов из сети Интернет, разрешая такие подключения только из ЛВС с ограниченного числа рабочих станций администраторов. Для этого необходимо применить соответствующие настройки на межсетевом экране. Кроме того, рекомендуется внедрить строгую парольную политику для исключения возможности установки простых или словарных паролей.

Если доступ к администрированию веб-сервера необходим, рекомендуется ограничить список IP-адресов, с которых такое подключение возможно, адресами рабочих станций администраторов.

Сценарий 2. Эксплуатация веб-уязвимостей

Чтобы получить возможность выполнять команды ОС, далеко не всегда требуется подбор учетных данных для доступа к интерфейсам управления. Зачастую подобную возможность дают уязвимости веб-приложений на сетевом периметре компании. Принимая во внимание тот факт, что веб-приложение в основном используется как публичный ресурс (официальный сайт, интернет-магазин, новостной портал и т. п.), доступ к нему должен быть обеспечен для любого пользователя сети Интернет. Это открывает множество возможностей для атак.

Среди наиболее опасных уязвимостей веб-приложений можно выделить загрузку произвольных файлов, внедрение операторов SQL и выполнение произвольного кода. Эксплуатация таких уязвимостей может привести к полной компрометации сервера.

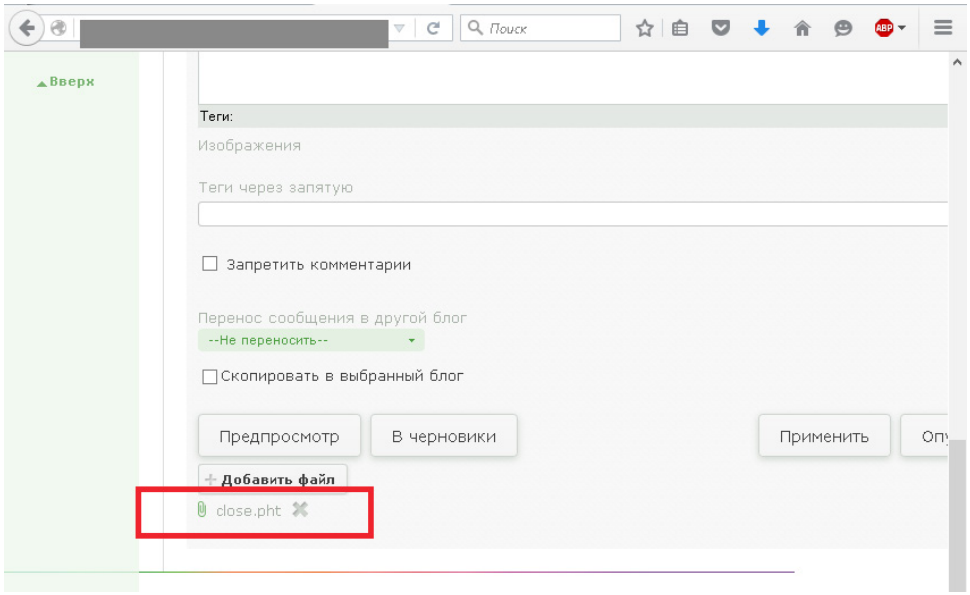
Вот пример наиболее простой для реализации атаки, которая успешно моделировалась в рамках наших тестов. В большинстве публичных веб-приложений существует возможность регистрации новых пользователей, а в личном кабинете таких пользователей, как правило, есть функция загрузки контента (фото, видео, документов, презентаций и др.). Обычно приложение проверяет, какой именно файл загружает пользователь, по списку запрещенных расширений — но зачастую эта проверка неэффективна. В таком случае злоумышленник может загрузить веб-интерпретатор командной строки на сервер, изменив расширение файла. В итоге нарушитель получит возможность выполнять команды ОС с привилегиями веб-приложения, а если эти привилегии были избыточны — то и полный контроль над ресурсом.

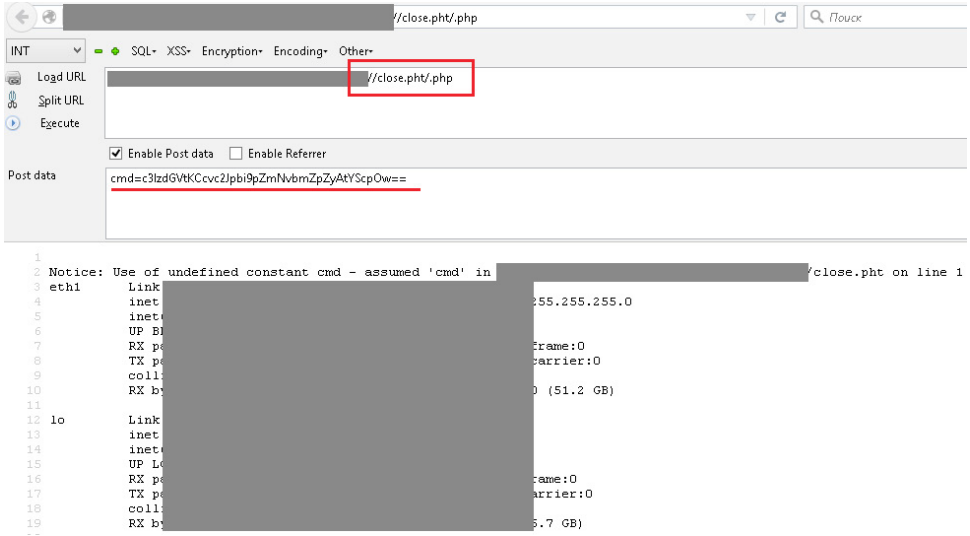
Даже если на сервере настроена эффективная проверка загружаемых файлов, необходимо учитывать и конфигурацию самой системы. Следующий пример демонстрирует, каким образом ошибка администратора может позволить нарушителю скомпрометировать ресурс.

В исследованном приложении была реализована проверка, которая запрещала загрузку файлов с расширением .php. Однако наши эксперты выяснили, что на сервере используется уязвимая комбинация ПО и ОС, которая позволяет обойти данное ограничение.

[7] https://msdn.microsoft.com/en-us/library/windows/desktop/aa378842(v=vs.85).aspx

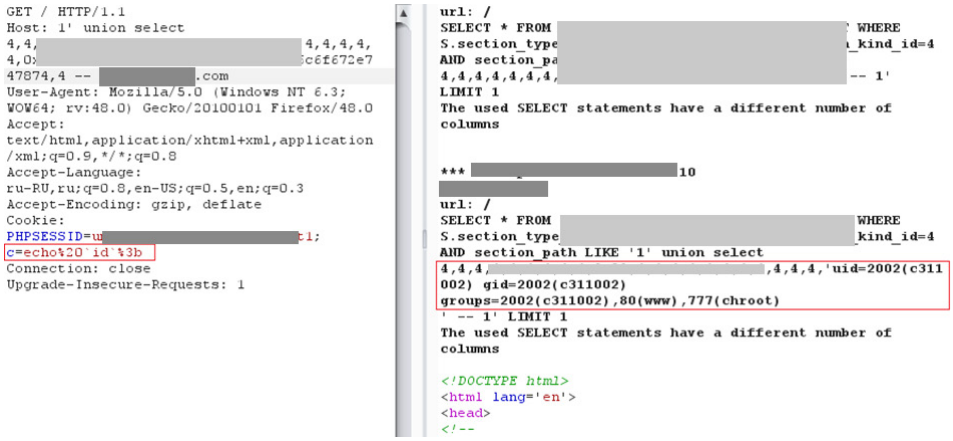
В частности, в конфигурации CMS Bitrix в файле /upload/.htaccess не было установлено ограничение на загрузку файлов с расширением .pht. Данный формат файла исполняется в ОС семейства Debian и Ubuntu как файл формата PHP. Таким образом, уязвимая конфигурация сервера позволила осуществить загрузку веб-интерпретатора командной строки на сервер в обход установленных ограничений.





Существуют и другие атаки на веб-приложения, которые позволяют выполнять команды на сервере — например, с помощью SQL-запроса. Как правило, для нарушителей такие атаки не составляют большого труда, однако назвать их тривиальными уже нельзя. На скриншоте ниже показан пример выполнения команды id через внедрение операторов SQL в комбинации с эксплуатацией уязвимости подключения локальных файлов.

Перечисленные методы подразумевают наличие у нарушителя определенных знаний о способах обхода фильтрации файлов при загрузке их на сервер или навыков написания SQL-запросов. Но такие знания могут и не потребоваться — например, если ограничения на загрузку файлов отсутствуют вовсе.

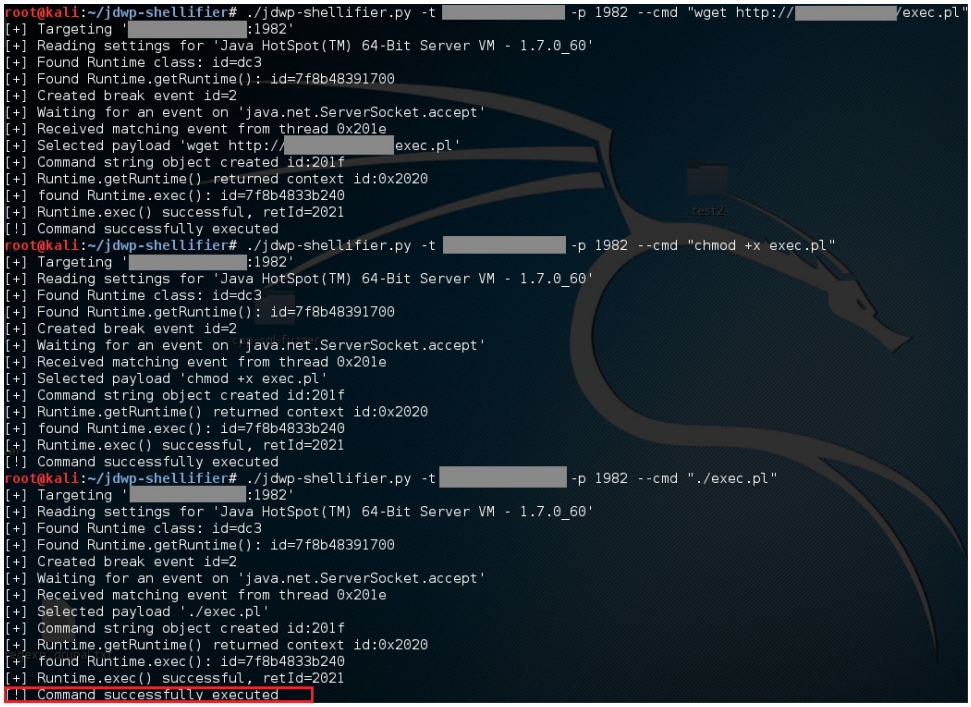


Рекомендации по защите. Помимо строгой парольной политики, рекомендуется выполнять проверку загружаемых на сервер файлов по методу белого списка. Для защиты от эксплуатации уязвимостей кода приложения (внедрение операторов SQL, выполнение команд и т. п.) необходимо реализовать фильтрацию передаваемых пользователем данных на уровне кода приложения. Кроме того, рекомендуется использовать межсетевой экран уровня приложения (web application firewall).

Необходимо также отметить, что в данном отчете показана лишь часть атак, которые могут быть реализованы в отношении веб-ресурсов. Больше деталей можно найти в специальных отчетах «Уязвимости веб-приложений»[8] и «Атаки на веб-приложения»[9].

Сценарий 3. Эксплуатация известных уязвимостей

Атаки на уязвимый протокол



[8] https://www.ptsecurity.com/upload/ptru/analytics/Web-Vulnerability-2016-rus.pdf
[9] https://www.ptsecurity.com/upload/corporate/ru-ru/analytics/Web-Applications-Attacks-rus.pdf

Еще одним примером использования недостатков фильтрации трафика на периметре сети является атака на протокол отладки Java Debug Wire Protocol (JDWP), один из компонентов системы Java Platform Debug Architecture (JPDA). Протокол не обеспечивает аутентификацию и шифрование трафика, чем могут воспользоваться внешние нарушители, если интерфейс JDWP доступен из Интернета. Злоумышленник может использовать общедоступный эксплойт[10] для выполнения команд ОС. Кроме того, служба, использующая JDWP, зачастую обладает максимальными привилегиями, что позволяет внешнему нарушителю за один шаг получить полный контроль над сервером. Ниже показан пример успешной атаки с использованием общедоступного эксплойта.

В рамках атаки на сервер был загружен файл exec.pl с backconnect-скриптом. Далее были изменены привилегии на исполнение этого файла. В результате запуска скрипта получен интерактивный шелл, который позволял выполнять команды ОС для развития атаки.

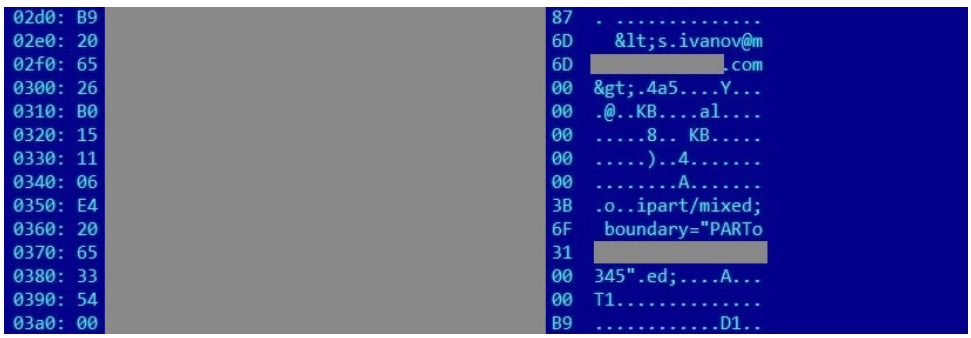
Рекомендации по защите. Данный пример показывает, как можно преодолеть периметр даже при использовании сложных паролей и регулярном обновлении ПО. Подобные отладочные интерфейсы не должны быть доступны из внешних сетей.

Атаки на уязвимое ПО

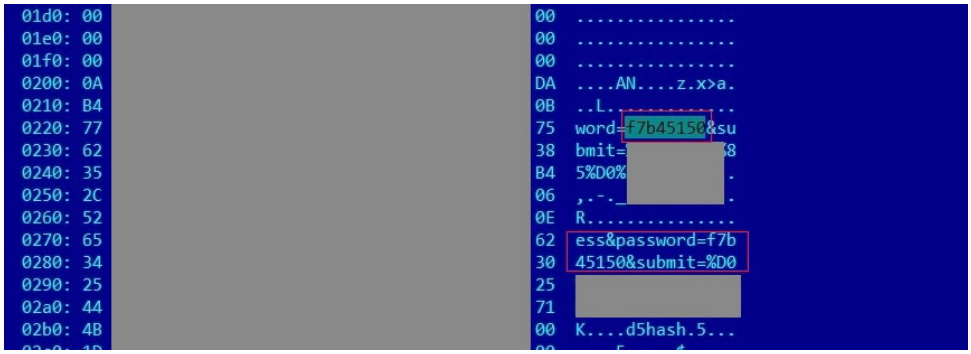
По статистике наших исследований, использование устаревших версий ПО — один из наиболее распространенных недостатков безопасности, выявляемых на сетевом периметре. Как правило, в рамках тестирований на проникновение эксплуатация уязвимостей ПО, позволяющих удаленно выполнять код, не производится, так как подобные атаки (например, направленные на переполнение буфера) могут вызвать отказ в обслуживании систем. Однако для потенциального нарушителя такое условие не является помехой. Более того, нарушение работы каких-либо компонентов КИС может быть его основной целью. Вот лишь некоторые примеры устаревших версий различных систем, часто встречаемых на сетевом периметре КИС, и их уязвимостей: Windows Server 2003 SP1, SP2 (CVE-2012-0002), Nginx 1.3.11 (CVE-2013-2028), PHP 5.3.8, 5.3.28, 5.5.1 и множества других версий (CVE-2014-3515, CVE-2011-3379, CVE-2013-6420), ProFTPD FTP Server 1.3.3a (CVE-2011-4130, CVE-2010-4221), OpenSSH Server 4.3 (CVE-2006-5051, CVE-2006-5052). До сих пор можно встретить даже использование ОС Windows XP с известной уязвимостью (CVE-2008-4250).

Часто эксплуатация таких уязвимостей требует от атакующего особых знаний и навыков, например для разработки собственного эксплойта. В то же время существуют и общедоступные, а также коммерческие эксплойты, которые могут быть использованы «из коробки» либо с минимальными изменениями для адаптации к конкретным условиям атаки.

В ряде проектов была продемонстрирована эксплуатация критически опасной уязвимости Heartbleed (CVE-2014-0160). Данная уязвимость, в случае поддержки сервисом



[10] https://github.com/IOActive/jdwp-shellifier



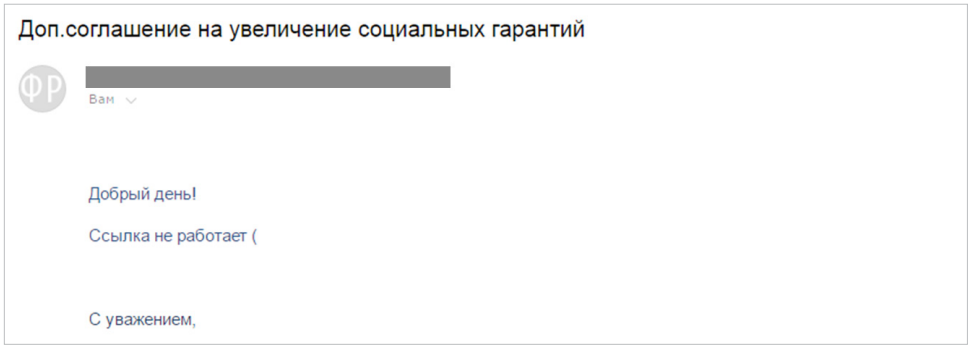
SSL-соединений, а также если операционная система узла принадлежит семейству *nix и на сервере установлена уязвимая версия библиотеки OpenSSL, — позволяет читать участки памяти серверного процесса (в данном примере веб-сервера). В таких участках памяти могут находиться в открытом виде критически важные данные: учетные данные пользователей, пользовательские сессии, ключи доступа и т. п. В результате проведения атаки и анализа участков памяти был, в частности, получен пароль пользователя.

Рекомендации по защите. Для предотвращения подобных атак рекомендуется своевременно обновлять используемое ПО и устанавливать обновления безопасности для ОС. Кроме того, желательно не раскрывать версии применяемых систем — в частности, версию веб-сервера, которая может указываться в стандартных сообщениях об ошибках или в HTTP-ответе.

Сценарий 4. Социальная инженерия

Социальная инженерия — один из наиболее распространенных методов целевых атак. Метод заключается в использовании недостаточной осведомленности сотрудников компании в вопросах безопасности. Нарушитель может выведать данные для доступа к ресурсам в телефонном разговоре, личной переписке.

Ниже описан пример социальной инженерии в телефонном разговоре с одним из сотрудников банка в рамках исследования по оценке осведомленности. Важно отметить, что данный сотрудник был выбран для разговора по результатам первичной рассылки фишинговых писем. Это был один из тех сотрудников, кто не просто перешел по ссылке из письма, но и вступил в переписку с экспертом Positive Technologies, приняв его за администратора своей корпоративной сети.

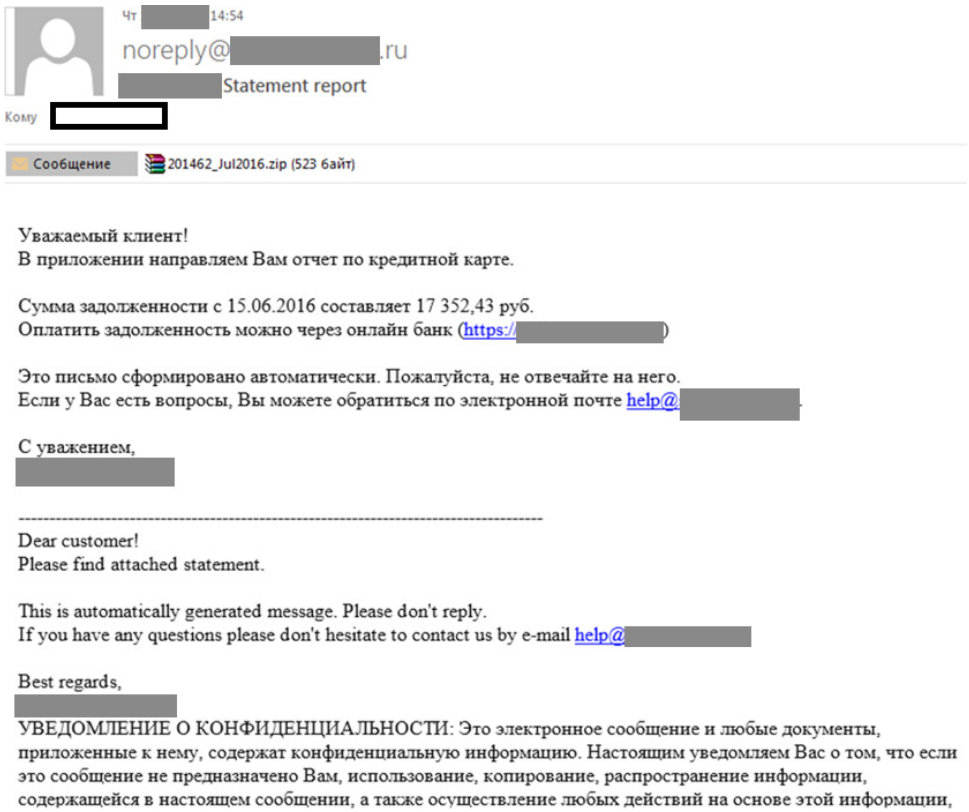


Наш эксперт представился администратором и предложил решить проблему неработающей ссылки в почтовой рассылке. Телефонный разговор с сотрудником длился около 4 минут, и этого времени оказалось достаточно, чтобы добиться поставленной цели — выведать учетные данные для доступа к рабочей станции сотрудника и ресурсам домена.

Сотрудник не только с легкостью выдал информацию об используемом ПО, но и свой пароль стороннему лицу, а также попросил не изменять его пароль, так как он «удобный» (то есть довольно простой). Потенциальный нарушитель мог не только получить доступ к рабочей станции и ресурсам домена от имени этого пользователя, но и быть уверенным в том, что сотрудник не сменит свой пароль и его можно будет использовать в течение длительного времени.

Естественно, не все люди так доверчивы, и при таком подходе велик риск, что сотрудник заподозрит неправомерные действия и обратится в службу безопасности свой компании. Поэтому часто используются более сложные социотехнические методы, требующие специальной подготовки, и сложность атаки возрастает.

Например, для использования фишинговых сценариев злоумышленник должен зарегистрировать собственный домен и разработать ложную форму аутентификации. Ему необходимо сделать фишинговый ресурс максимально приближенным по дизайну страницы к тому ресурсу, которым привык пользоваться сотрудник. Атакующий также разрабатывает специальные сценарии для определения версий ПО, используемого сотрудником, и сценарии последующей эксплуатации уязвимостей этого ПО. Если нарушитель ставит целью заразить рабочую станцию вредоносным ПО, ему необходимо учитывать, какие системы защиты используются на узле, для этого требуются дополнительные разведывательные действия. Все это существенно повышает сложность реализации успешной атаки. Однако, как показывает опыт наших тестирований на проникновение и расследований реально произошедших инцидентов ИБ, социотехнические атаки могут быть успешно реализованы в большинстве современных организаций. Именно такие атаки являются в последние годы первым шагом киберпреступников по проникновению в КИС банков, государственных организаций и промышленных корпораций.



Чт [redacted] 14:54
noreply@[redacted].ru
[redacted] Statement report
Кому [redacted]
Сообщение 201462_Jul2016.zip (523 байт)

Уважаемый клиент!
В приложении направляем Вам отчет по кредитной карте.

Сумма задолженности с 15.06.2016 составляет 17 352,43 руб.
Оплатить задолженность можно через онлайн банк (https://[redacted])

Это письмо сформировано автоматически. Пожалуйста, не отвечайте на него.
Если у Вас есть вопросы, Вы можете обратиться по электронной почте help@[redacted].

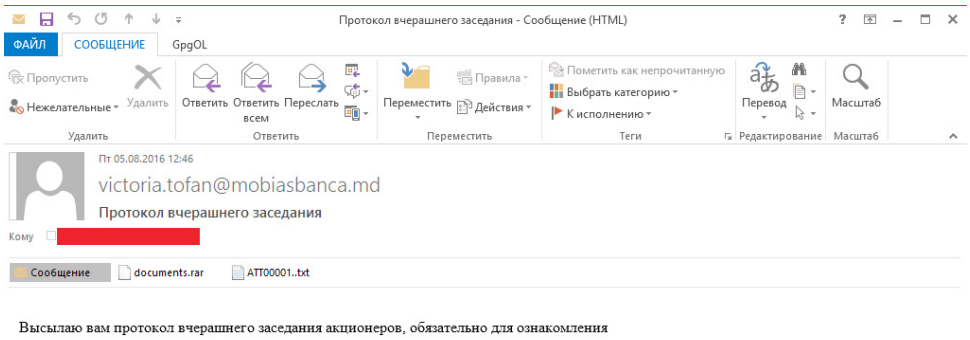
С уважением,
[redacted]

Dear customer!
Please find attached statement.

This is automatically generated message. Please don't reply.
If you have any questions please don't hesitate to contact us by e-mail help@[redacted]

Best regards,
[redacted]
УВЕДОМЛЕНИЕ О КОНФИДЕНЦИАЛЬНОСТИ: Это электронное сообщение и любые документы, приложенные к нему, содержат конфиденциальную информацию. Настоящим уведомляем Вас о том, что если это сообщение не предназначено Вам, использование, копирование, распространение информации, содержащейся в настоящем сообщении, а также осуществление любых действий на основе этой информации,

Выше приведен пример фишингового письма, которое использовалось специалистами Positive Technologies в рамках нескольких тестирований на проникновение в различных организациях в 2016 году. В этом письме используется домен, который по написанию схож с реально существующим. Внимательный сотрудник может легко обнаружить подозрительный адрес отправителя — однако, как показывает практика, далеко не все сотрудники замечают подмену. Кроме того, нарушитель может изменить адрес отправителя на реально существующий адрес одного из сотрудников компании, чтобы не вызвать подозрений. Загрузка приложенного файла и попытка распаковки архива в рамках тестирования на проникновение приводили лишь к отправке информации о пользователе и используемом им ПО на адрес проверяющих. Однако в случае реальной атаки рабочая станция жертвы могла быть сразу заражена вредоносным ПО.



Во время одного из расследований инцидентов ИБ наши эксперты выявили аналогичный вектор проникновения в ЛВС банка с помощью вредоносного ПО. «Вредонос» тоже был разослан по электронной почте в архиве, при этом рассылка фишинговых писем осуществлялась с адресов сотрудников партнерского банка, с которыми жертвы привыкли вести переписку в рамках рабочего процесса. Адреса были подделаны злоумышленниками, которые предварительно провели разведку и изучили специфику почтовой переписки сотрудников (вероятно, атакам подвергся и партнерский банк).

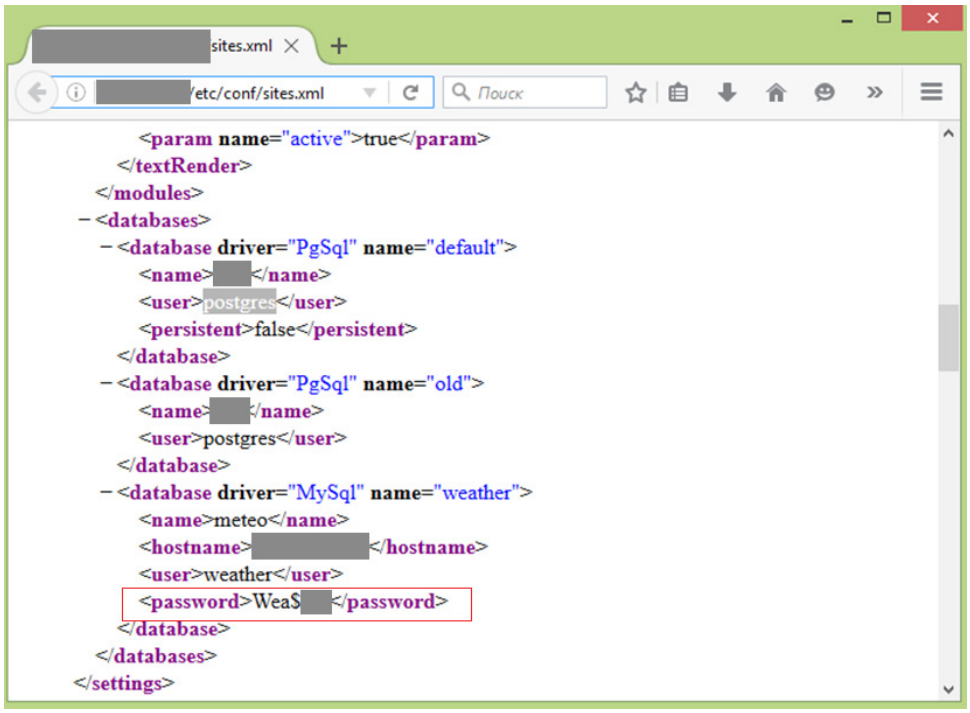
Рекомендации по защите. Приведенные примеры свидетельствуют о том, что некоторые атаки методами социальной инженерии могут быть легко выявлены сотрудниками, необходимо лишь быть более бдительными — всегда проверять адрес отправителя, не переходить по сомнительным ссылкам и не запускать приложенные к письму файлы, если нет уверенности в безопасности их содержимого. Кроме того, ни при каких обстоятельствах нельзя сообщать никому свои учетные данные, в том числе администраторам и сотрудникам службы безопасности.

Однако существуют методы атак, которые выявить крайне трудно, например если письма приходят от доверенного лица. Для защиты от таких атак рекомендуется использовать антивирусные решения, способные проверять файлы, получаемые по электронной почте, до открытия их сотрудником. Некоторые антивирусы позволяют также помещать сомнительные файлы в так называемую песочницу и в этой безопасной среде исследовать активность вредоносного ПО. В любом случае рекомендуется регулярно проводить тренинги для сотрудников компании с целью повышения их осведомленности в вопросах ИБ, а также оценивать эффективность таких тренингов в рамках внутренних проверок и с помощью тестирования на проникновение методами социальной инженерии.

Сценарий 5. Открытые данные

Этот метод сам по себе не является атакой, однако в рамках многих тестирований на проникновение эксперты Positive Technologies используют его для успешного преодоления периметра, как минимум в качестве первого шага при реализации других атак.

Исследование страниц веб-приложений зачастую позволяет выявить множество ценной информации в открытом виде — учетные записи пользователей, версии ПО и серверов, адреса критически важных систем, конфигурационные файлы оборудования и даже исходный код веб-приложения. Любой внешний нарушитель может получить доступ к ресурсам с возможностью загрузки произвольных файлов без каких-либо атак на систему, если выявит учетную запись, например для доступа к ресурсу по протоколу SSH, для подключения к СУБД или к интерфейсу администрирования веб-приложения.



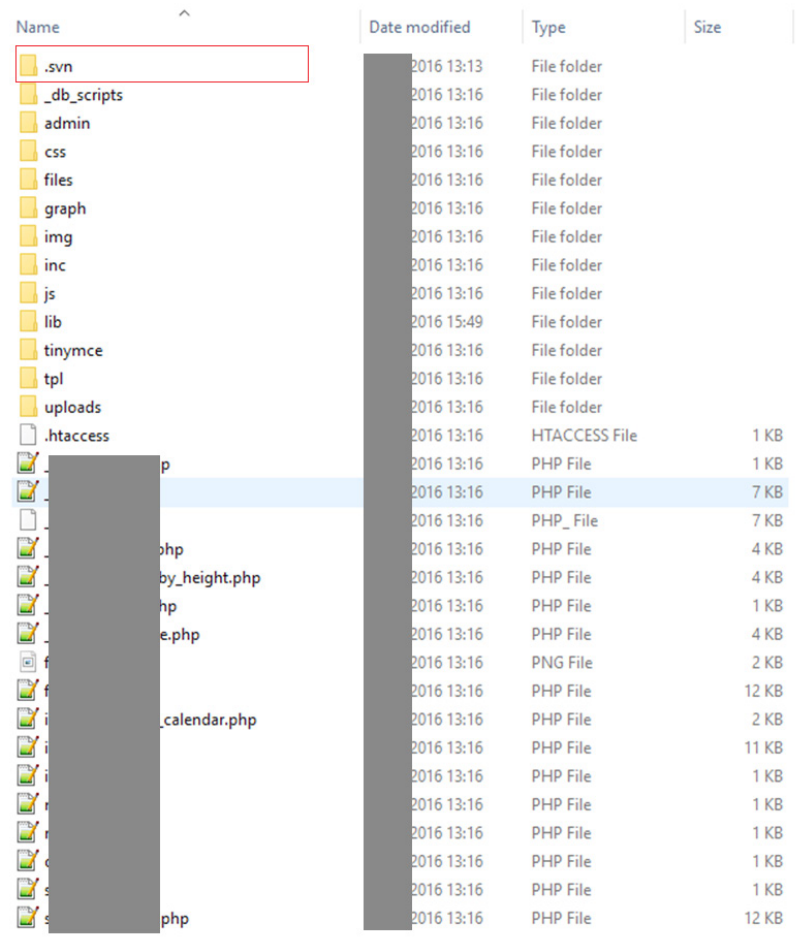
Также в открытом доступе может быть обнаружена и доменная учетная запись. В рамках одного из тестирований на проникновение именно это позволило нашим экспертам получить доступ к беспроводной сети, из которой был возможен доступ к контроллерам доменов в ЛВС. В другом проекте такая учетная запись позволила получить доступ к множеству корпоративных ресурсов компании, доступных из Интернета, в частности к системе Jira (развитие данного вектора атаки описано в сценарии 6).

Следующий пример показывает, как злоумышленник может использовать исходный код приложения. В данном примере в открытом доступе на периметре сети были обнаружены файлы директории .svn. Для получения исходного кода внешний нарушитель может использовать ПО dvcs-ripper и Subversion.

Если внешнее тестирование на проникновение подразумевает моделирование действий злоумышленника, проводящего атаки без каких-либо дополнительных знаний об атакуемой КИС (то есть методом черного ящика), то в случае получения исходного кода приложения нарушитель сможет проводить анализ методом белого ящика, то есть обладая полным набором сведений о приложении. Для анализа кода могут использоваться

как ручные средства, так и широкодоступные автоматизированные решения. Все это позволяет выявить максимально возможное число уязвимостей и подготовить эксплойты для атаки.

```
root@kali:/# dirb https://[redacted].com/

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: [redacted] 15:32:58 2016
URL_BASE: https://[redacted].com/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: https://[redacted].com/
==> DIRECTORY: https://[redacted].com/.svn/
+ https://[redacted].com/.svn/entries (CODE:200|SIZE:3)
+ https://[redacted].com/_ (CODE:200|SIZE:32793)
+ https://[redacted].com/_data (CODE:200|SIZE:51509)
+ https://[redacted].com/0 (CODE:200|SIZE:32638)
```



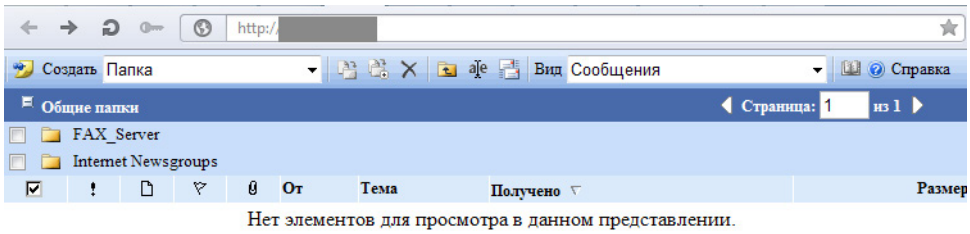
При анализе полученных файлов было выявлено, что в одном из них в открытом виде хранится учетная запись администратора веб-приложения. Кроме того, были выявлены уязвимости, позволяющие читать и загружать файлы на север — а это дает возможность получить полный контроль над ресурсом, как было продемонстрировано в описании предыдущих сценариев.

```
CREATE TABLE `users` (
  `user_id` int(10) UNSIGNED NOT NULL,
  `user_login` varchar(50) NOT NULL,
  `user_pass` varchar(50) NOT NULL,
  `user_pass_date` date NOT NULL,
  `user_name` varchar(50) NOT NULL,
  `user_email` varchar(50) NOT NULL,
  `user_admin` tinyint(1) UNSIGNED NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8;

--
-- Dumping data for table `users`
--

INSERT INTO `users` (`user_id`, `user_login`, `user_pass`, `user_pass_date`, `user_name`, `us
(1, '[redacted]', '123123', '2015-11-27', '[redacted]', 'info@[redacted]ru', 1),
(2, '[redacted]', '[redacted]gfhjkm', '2014-09-11', '[redacted]', '[redacted].com', 1),
(4, 'Ilya', '123456', '2014-09-12', 'Ilya [redacted]', '[redacted].com', 0),
(5, '[redacted]', '123123', '2015-02-10', 'Anna [redacted]', '[redacted].com', 0),
(6, 'test', '123456', '2015-02-05', 'Test', 'test@test.com', 0),
(7, 'test2', '123456', '2015-04-08', 'Test2', 'test2@test.com', 0);
```

Рекомендации по защите. Администраторам систем необходимо следить за тем, какие данные раскрываются на страницах веб-ресурсов, и обеспечивать эффективное разграничение доступа к файлам и директориям на серверах, доступных из внешних сетей.

Сценарий 6. Выход из песочницы

На сетевом периметре КИС, как правило, располагаются публичные сервисы организации — веб-приложения, доступные по протоколам HTTP и HTTPS. Однако некоторые компании располагают на периметре и корпоративные ресурсы, почтовые сервисы (OWA), порталы и другие системы.



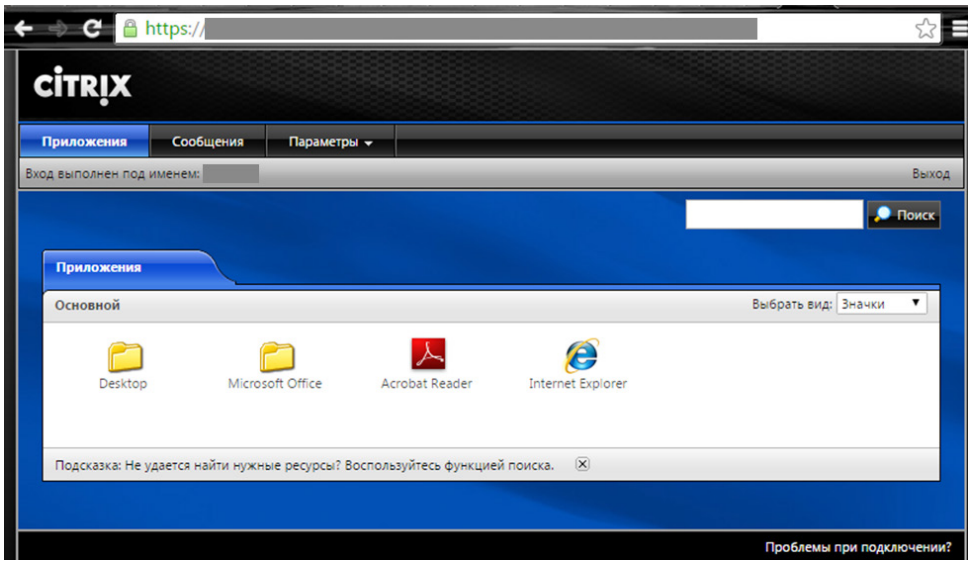
Рассмотрим сценарий атаки, которая начиналась с получения доменной учетной записи в открытом виде с общедоступной страницы веб-приложения, что позволяло подключиться ко множеству корпоративных ресурсов на периметре сети (см. сценарий 5).

Среди таких ресурсов была система Jira, при подключении к которой внешний нарушитель может получить список всех пользователей домена. Эксперты Positive Technologies выгрузили такой список и подобрали пароль P@ssw0rd к учетной записи одного из доменных пользователей. (Кстати, этот пароль является одним из наиболее распространенных для ресурсов КИС[11], так что теоретически эта учетная запись могла быть подобрана напрямую — например, если подбирать для данного пароля различные имена пользователей; именно такой метод используется при тестировании на проникновение для подбора доменных учетных записей, во избежание блокировки. Данный метод не входит в описываемый сценарий атаки, но еще раз показывает, насколько важно уделять внимание парольной политике и безопасному хранению учетных данных).

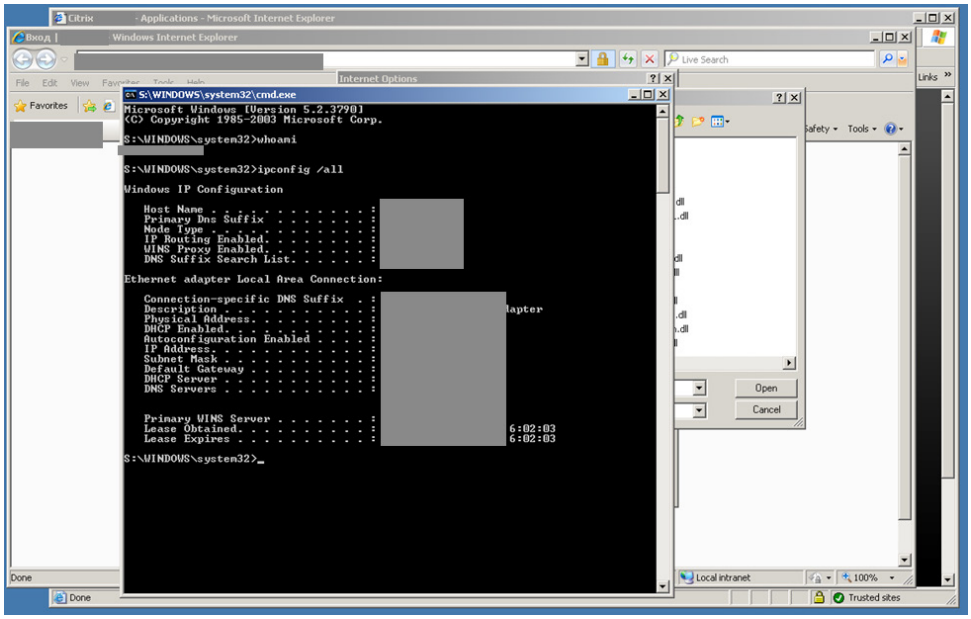
Подобранная учетная запись была использована для подключения к еще одному из корпоративных ресурсов компании, доступных на сетевом периметре, — системе Citrix. Атака на эту систему и отражает суть сценария 6.

[11] https://www.ptsecurity.com/upload/ptru/analytics/Corporate-Vulnerability-2015-rus.pdf

Citrix используется для виртуализации и обеспечивает удаленный доступ к корпоративным приложениям и рабочим столам рабочих станций и серверов с любого устройства. Обладая доступом к такой системе, пользователь не должен получать возможность выхода из системы виртуализации и выполнения команд ОС непосредственно на сервере, где система Citrix установлена. Однако существуют методы обхода песочницы, которыми часто пользуются нарушители.



К примеру, запустив в Citrix браузер Internet Explorer, нарушитель может использовать встроенную функцию — открытие файла. Если на сервере не настроено строгое разграничение доступа к файлам и директориям, эта функция браузера позволяет получить доступ к файловой системе, в том числе к директории установки ОС, и запустить файл cmd.exe для выполнения произвольных команд. Аналогичный вектор атаки можно реализовать и с помощью другого ПО, где доступна функция открытия файла.



Рекомендации по защите. В данном примере показана эксплуатация уязвимости, связанной с недостаточно эффективным разграничением доступа к функциям и файлам ОС.

Используя встроенные функции прикладного ПО, нарушитель может получить доступ к любым файлам на сервере. Это серьезная ошибка администрирования ресурса.

Для предотвращения подобных атак следует пересмотреть вопрос о необходимости размещения корпоративных ресурсов на периметре сети. Если такое размещение действительно необходимо — реализовать строгую парольную политику, а также строгое разграничение доступа к директориям и файлам ОС, чтобы пользователи таких систем, как Citrix, не могли получить доступ к файловой системе сервера, особенно на исполнение файлов (в частности, ограничить доступ к директории установки ОС). При реализации разграничения доступа необходимо придерживаться принципа минимизации привилегий. Кроме того, рекомендуется использовать защищенный протокол TLS с проверкой наличия корневого сертификата на клиенте для запуска ПО в системе Citrix.

Получение контроля над КИС

Атаки на ресурсы внутренних сетей КИС, как правило, осуществляются от лица двух типов нарушителей — внутреннего, обладающего доступом к сетевой розетке на территории организации, либо внешнего, успешно преодолевшего сетевой периметр. Модель внутреннего нарушителя может меняться в зависимости от того, из какого сегмента сети развивается вектор атаки, а также в зависимости от уровня начальных привилегий атакующего.

Если для реализации атак на ЛВС со стороны Интернета не требуется проходить дополнительную аутентификацию в сети (так как нарушитель уже получил определенный уровень привилегий на скомпрометированном сервере, находящемся в определенном сетевом сегменте), то внутреннему нарушителю необходимо каким-либо образом получить логический доступ к ЛВС и привилегии на каком-либо внутреннем ресурсе — если, конечно, нарушитель не является сотрудником организации, который уже обладает подобными привилегиями.

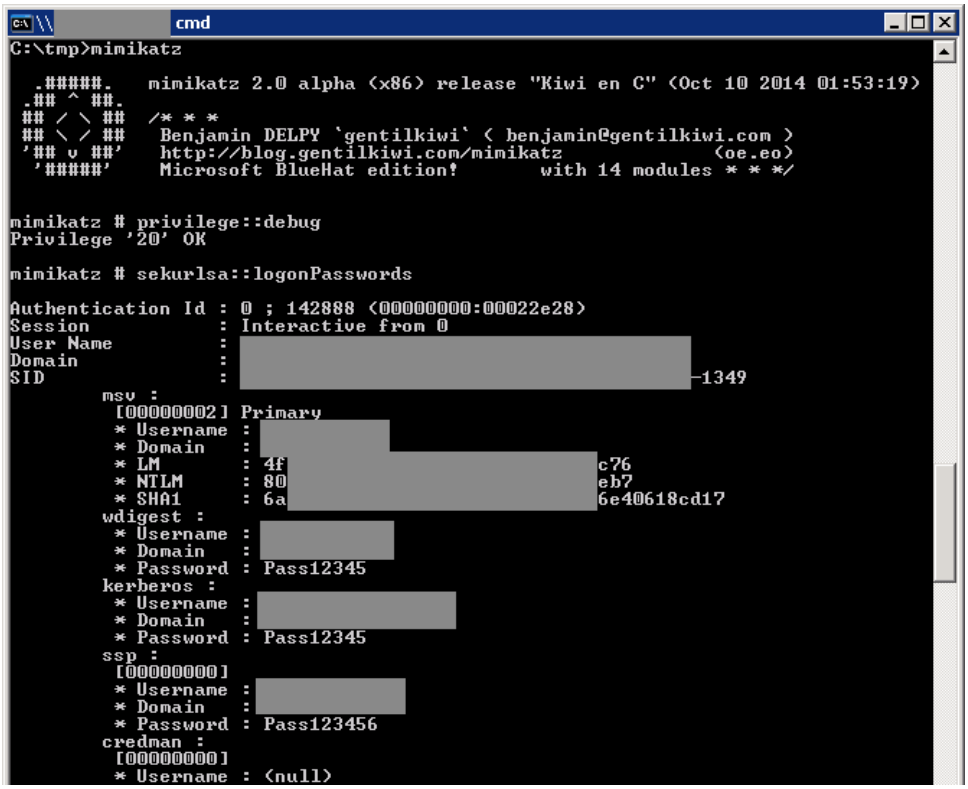
Сложность развития атак со стороны внутреннего нарушителя во многом определяется конфигурацией сети и сетевого оборудования. В первую очередь — сегментацией, фильтрацией сетевых протоколов, а также настройками защиты сети от подключения стороннего оборудования. К сожалению, далеко не все организации обеспечивают достаточный уровень защиты КИС на уровне сети.

Как правило, КИС современных организаций построена на базе доменов (Microsoft Active Directory). Подобная реализация удобна и способна обеспечивать централизованное управление даже крупными распределенными сетевыми инфраструктурами. Однако она может быть уязвима для атак в случае недостаточного внимания к обеспечению безопасности со стороны как администраторов, так и рядовых пользователей КИС. Как показывает практика, наиболее распространенными проблемами в КИС такого рода являются недостаточная строгость парольной политики и недостаточная защита привилегированных учетных записей домена.

Самым простым и самым распространенным сценарием атаки на КИС, построенную на базе Microsoft Active Directory, является комбинация двух несложных действий внутреннего нарушителя — получения привилегий локального администратора на узле ЛВС и запуска специализированных утилит для взлома на скомпрометированном ресурсе с целью получения учетных данных пользователей.

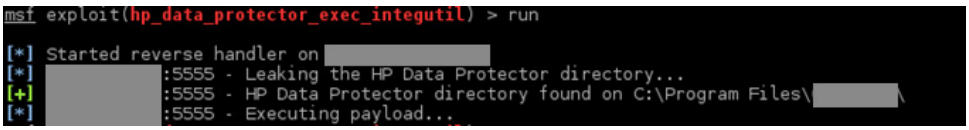
Учетная запись локального администратора может быть использована для получения паролей в открытом виде. Это возможно из-за слабости архитектурной реализации single sign-on (SSO) во всех системах Windows, поддерживающих этот механизм. Уязвимость

существует из-за того, что подсистемы Windows wdigest, kerberos и tspkg хранят пароли пользователей с помощью обратимого кодирования в памяти операционной системы. Таким образом, локальный администратор имеет возможность получить доступ к паролям всех пользователей, авторизованных в системе. Для реализации подобных атак существует специальный инструментарий, который можно найти в Интернете в свободном доступе (например, утилиты Mimikatz или WCE).



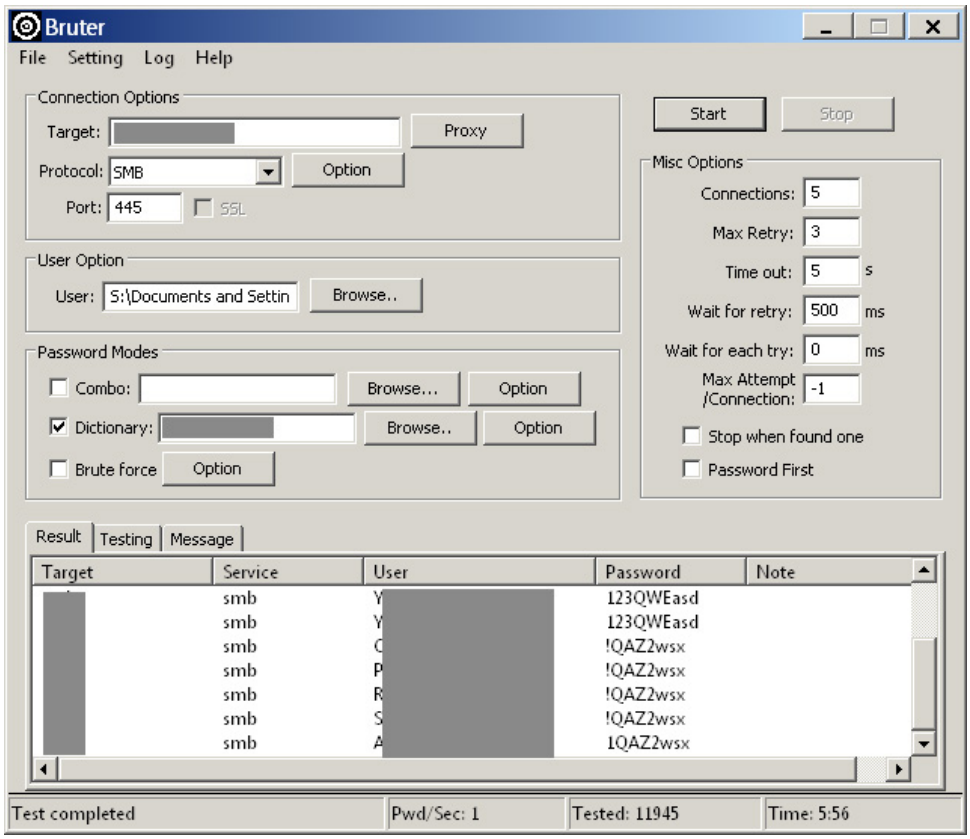
Повторяя эти шаги последовательно на ряде узлов ЛВС, нарушитель может добраться до того ресурса, на котором активна учетная запись администратора домена, и получить ее в открытом виде.

Описанные ниже сценарии 1 и 2 по сути различаются лишь методом получения привилегий локального администратора на первом шаге. Всего же в данном отчете выделено 7 сценариев атаки, которые наиболее часто встречаются в проектах по внутреннему тестированию на проникновение, — то есть атаки злоумышленников с использованием этих техник наиболее вероятны. Восьмым сценарием можно считать эксплуатацию известных уязвимостей ПО и ОС, но эти сценарии менее интересны с точки зрения техники эксплуатации уязвимости (например, использование общедоступного эксплойта, как показано на рисунке ниже), и в данном отчете они подробно рассматриваться не будут.



Сценарий 1. Подбор доменной учетной записи

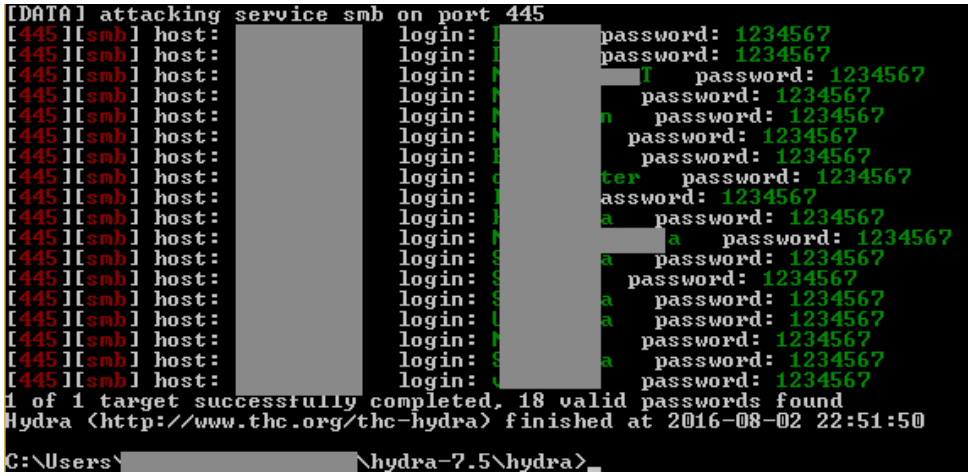
В большинстве КИС настроены определенные парольные политики для доменных учетных записей, однако далеко не во всех организациях такие политики эффективны. Зачастую ограничения позволяют задавать словарные комбинации. Примером наиболее распространенного пароля, удовлетворяющего большинству применяемых парольных политик, можно считать P@ssw0rd. Данная комбинация обладает достаточной длиной и сложностью для того, чтобы считать ее стойкой к подбору, однако она включена в большинство словарей наиболее популярных паролей и наверняка будет проверена нарушителем в первую очередь. Пример атаки с использованием такого пароля описан выше, в разделе о методах преодоления периметра КИС по сценарию 6. Подобрать удается и более сложные комбинации — с помощью словарей часто используемых паролей.



Как правило, в доменной инфраструктуре настроены ограничения на количество попыток ввода неверного пароля, с последующей блокировкой учетной записи для защиты от автоматизированных атак на учетные данные. Однако нарушитель может осуществлять подбор одного (или двух) паролей для целого списка идентификаторов пользователей — если у него есть информация о них. Получить такие данные несложно: внутреннему нарушителю (сотруднику организации) достаточно сделать запрос к контроллеру домена либо проанализировать адресную книгу почтового клиента; внешний же нарушитель может изучить открытые источники в Интернете (публикации компании, презентации, контактные данные с официального сайта), а также использовать недостатки защиты при хранении чувствительных данных на внешних ресурсах организации.

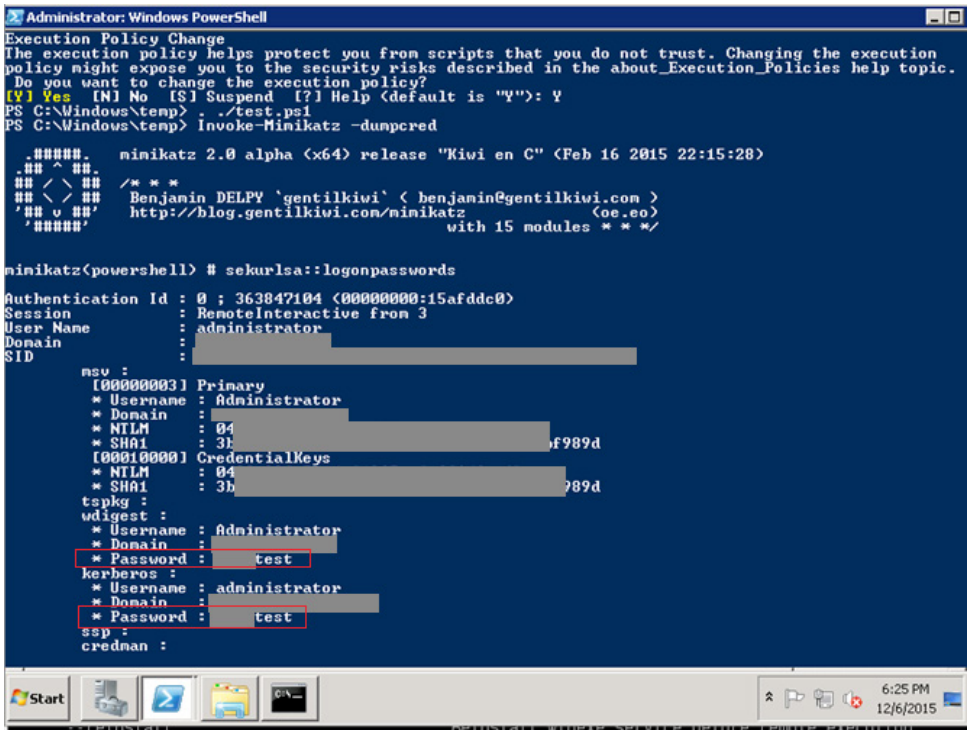
Кроме того, изучив принцип составления идентификатора доменной учетной записи, нарушитель может составить словарь для подбора. Наиболее распространенным принципом составления идентификатора является комбинация первой буквы имени с фамилией

сотрудника (например, DOMAIN\AIvanov), часто встречаются добавление первой буквы отчества (DOMAIN\APIvanov) и прочие вариации, основанные на ФИО.



Подобранная учетная запись нередко обладает привилегиями локального администратора на одной из рабочих станций или на сервере ЛВС, что позволяет нарушителю подключиться к данному узлу удаленно (например, по протоколу RDP) и запустить ПО для взлома.

Основной преградой на пути нарушителя в данном случае может стать антивирусное ПО, но часто оно бывает недостаточно эффективно настроено, чтобы противостоять атакам. В практике наших тестирований на проникновение проблема эффективности антивирусной защиты на узлах ЛВС встречается практически в каждом проекте. Либо антивирусное ПО вовсе не запрещает запуск используемых при атаке утилит, либо привилегии локального администратора позволяют отключить антивирус или добавить ПО в список исключений.



Но даже в случае блокировки утилит антивирусной системой и невозможности ее отключения опытный злоумышленник сможет провести атаку. Для обхода защиты нарушителю достаточно запустить утилиту с любого общедоступного ресурса в ЛВС либо сделать дамп памяти процесса lsass.exe (например, утилитой procdump) и запустить утилиту Mimikatz уже на собственной рабочей станции. Кроме того, существует реализация этой утилиты на языке PowerShell, которая не определяется антивирусными системами как опасное ПО.

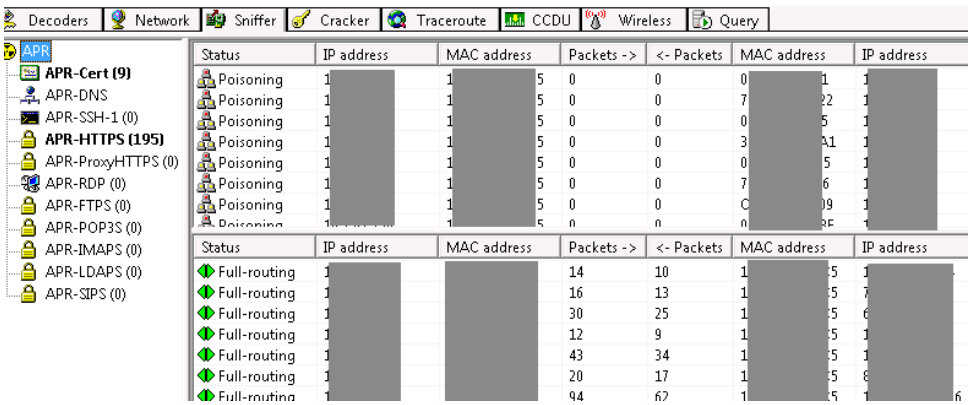
В результате подобной атаки нарушитель получает учетные данные всех пользователей, которые аутентифицировались в ОС, в открытом виде. Среди таких пользователей могут быть как локальные администраторы других узлов, так и привилегированные учетные записи домена. Подобный вектор атаки используется для получения полного контроля над доменной инфраструктурой.

Рекомендации по защите. Данный сценарий практически в каждом случае завершается успешно. Минимизировать риск подобной атаки можно лишь за счет строгой парольной политики, предотвращающей использование простых паролей всеми без исключения пользователями домена, а также за счет ограничения привилегий локальных пользователей на рабочих станциях и серверах домена. Для привилегированных учетных записей рекомендуется использовать двухфакторную аутентификацию. При этом важно понимать, что двухфакторная аутентификация также подвержена атакам (см. ниже сценарий 5).

Сценарий 2. Атаки на протоколы сетевого и канального уровней

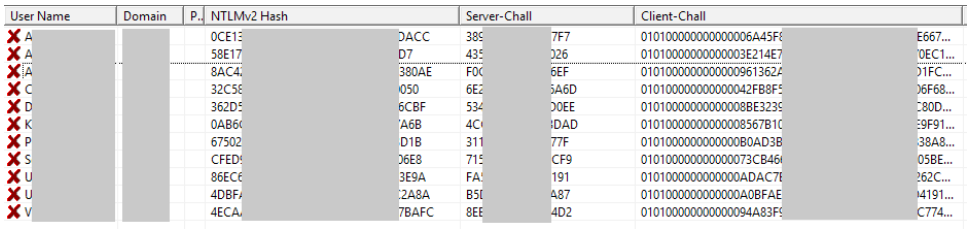
Если подобрать учетные данные не удалось либо у нарушителя отсутствует список идентификаторов пользователей домена, злоумышленник может анализировать используемые в ЛВС протоколы. В частности, он может проводить атаки методом «человек посередине» с целью перехвата трафика (например, если удастся реализовать атаку ARP Poisoning) либо атаки на протоколы NBNS и LLMNR с целью перехвата идентификаторов и хешей паролей пользователей.

Атака ARP Poisoning хорошо известна, поэтому в данном отчете основное внимание уделено атакам на другие протоколы. Кроме того, в рамках тестирований на проникновение атаки на протокол ARP осуществляются только по согласованию с заказчиком услуги, который, как правило, против такой демонстрации из-за высокой вероятности нарушения работы сети.



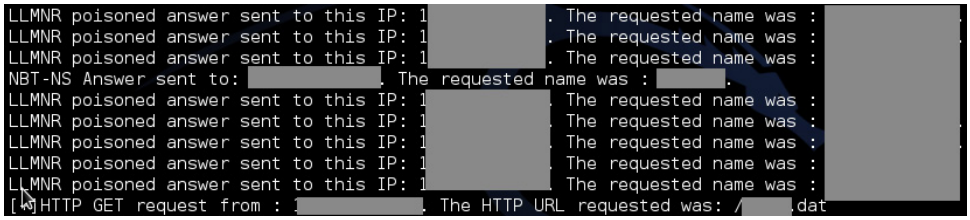
В результате атаки «человек посередине», в частности, могут быть перехвачены значения Challenge-Response для пользователей домена. По полученным значениям

Challenge-Response возможен подбор пароля пользователя. Причем подбор может осуществляться уже без доступа к системе, на ресурсах нарушителя.



Протоколы NetBIOS Name Service (NBNS) и Link Local Multicast Name Resolution (LLMNR) используются для получения IP-адреса узла в том случае, если такая запись отсутствует на DNS-серверах, или в случае, если эти серверы по тем или иным причинам недоступны. Отсутствие механизмов защиты этих протоколов чревато проведением атак LLMNR Spoofing и NBNS Spoofing.

Нарушитель, находящийся в одном сегменте сети с атакуемым узлом, может прослушать широковещательный трафик данных протоколов и подменить узел, на котором целевой узел осуществляет попытку аутентификации. В случае успешной атаки злоумышленник сможет прослушивать и модифицировать трафик в сетевом сегменте, а также получать аутентификационные данные пользователей и использовать их для доступа к другим узлам сети.



Получив идентификаторы и хеши паролей пользователей в результате подобных атак, злоумышленник может подобрать пароли по значениям хешей. Кроме того, нарушитель получает возможность использовать идентификаторы пользователей для развития атаки по сценарию 1.

Завершающий этап атаки (в случае успешного подбора учетных данных) аналогичен сценарию 1 — подключение к узлам, на которых полученная учетная запись обладает привилегиями локального администратора, и последующий запуск специализированных утилит для взлома.

Рекомендации по защите. При отсутствии необходимости в таких протоколах их следует отключать, а в случае необходимости их использования — применять превентивные меры защиты (например, объединять системы, использующие такой протокол, в отдельные сегменты сети). Методы защиты от атак ARP Poisoning хорошо известны: использовать статические ARP-записи на шлюзах, задействовать функции систем обнаружения атак

(например, препроцессора arpspoof системы Snort[12]) или утилиты, такие как arpwatch[13], задействовать функции Dynamic ARP Inspection коммутаторов Cisco и другие.

Сценарий 3. Атака SMB Relay

Использование в сети протоколов NBNS и LLMNR позволяет не только проводить атаки с целью перехвата хешей паролей пользователей, но и всем известную атаку SMB Relay. Этот метод атаки позволяет нарушителю перехватить аутентификационные данные, передаваемые от одного узла к другому, в процессе обмена информацией NTLM Challenge-Response. Принцип атаки прост: нарушитель слушает сетевой трафик и ждет, когда какой-либо узел или автоматизированная система инициирует подключение к другому узлу. Как только такой запрос обнаружен, нарушитель реализует атаку «человек посередине» (например, LLMNR Spoofing), перехватывает запрос на аутентификацию от обратившегося узла и передает его на атакуемый сервер. Этот сервер возвращает ответ (просьбу зашифровать некоторое сообщение с помощью своего хеша), такой ответ аналогичным образом перенаправляется на узел, запросивший подключение. На следующей итерации аналогичным образом происходит перенаправление этого зашифрованного сообщения. Так как сообщение было зашифровано корректным хешем, атакуемый сервер отправит нарушителю разрешение на аутентификацию. Злоумышленник аутентифицируется на сервере, а узлу, запросившему аутентификацию, отправит ответ об ошибке подключения. Более того, нарушитель может реализовать такую атаку не в отношении какого-то третьего узла, но и в отношении того же ресурса, который отправляет запрос на подключение.

Данная атака известна очень давно, компания Microsoft выпустила бюллетень безопасности MS08-068[14] и соответствующее обновление безопасности для ОС еще в 2008 году. Если этот патч установлен на узле, то нарушитель не сможет провести атаку на этот же узел, если он инициирует подключение. Но возможность атаковать с помощью SMB Relay другие узлы доменной инфраструктуры останется (если на них не реализована подпись SMB-пакетов — SMB Signing).

```
[+] HTTP Options:
    Always serving EXE          [OFF]
    Serving EXE                 [OFF]
    Serving HTML                [OFF]
    Upstream Proxy              [OFF]

[+] Poisoning Options:
    Analyze Mode                [OFF]
    Force WPAD auth             [OFF]
    Force Basic Auth            [OFF]
    Force LM downgrade          [OFF]
    Fingerprint hosts           [ON]

[+] Generic Options:
    Responder NIC               [eth0]
    Responder IP                [10.      1]
    Challenge set               [1122334455667788]
    Respond To                  ['10.      4']
    Don't Respond To Names      ['      ']


[+] Listening for events...
[*] [LLMNR]  Poisoned answer sent to 10.      4 for name
[FINGER] OS Version     : Windows 7 Professional 7601 Service Pack 1
[FINGER] Client Version : Windows 7 Professional 6.1
[*] [LLMNR]  Poisoned answer sent to 10.      4 for name
[FINGER] OS Version     : Windows 7 Professional 7601 Service Pack 1
[FINGER] Client Version : Windows 7 Professional 6.1
[*] [LLMNR]  Poisoned answer sent to 10.      4 for name
[FINGER] OS Version     : Windows 7 Professional 7601 Service Pack 1
[FINGER] Client Version : Windows 7 Professional 6.1
[+] Exiting...
```

[12] http://www.snort.org/
[13] http://xgu.ru/wiki/man:arpwatch
[14] https://technet.microsoft.com/library/security/ms08-068

Простоту реализации атаки покажем на примере одного из проектов по тестированию на проникновение. Анализируя трафик сети, мы выявили, что один из узлов периодически запрашивает адрес другого узла, после чего делает на него HTTP-запрос с доменной учетной записью. С помощью утилиты Responder была реализована успешная атака на выбранный нами узел сети с использованием запроса на подключение с того узла, который изначально инициировал запрос на подключение.

Выполнение команд на атакованном сервере возможно с привилегиями того пользователя, чьи аутентификационные данные были перехвачены в рамках SMB Relay (в нашем случае привилегии оказались максимальными). В результате был получен полный контроль над сервером.

```
SMB signing: False
Os version: 'Windows Server 2008 R2 Standard 7601 Service Pack 1'
Hostname: '[redacted]'
Part of the [redacted] domain
[+] Setting up HTTP relay with SMB challenge: 5b[redacted]30
[+] Received NTLMv2 hash from: 10.[redacted]4
[+] Client info: ['Windows 7 Professional 7601 Service Pack 1', domain:[redacted] signing:'False']
[+] Username: serveradmin is whitelisted, fowarding credentials.
[+] SMB Session Auth sent.
[+] Looks good, serveradmin has admin rights on C$.
[+] Authenticated.
[+] Dropping into Responder's interactive shell, type "exit" to terminate

Available commands:
dump                -> Extract the SAM database and print hashes.
regdump KEY         -> Dump an HKLM registry key (eg: regdump SYSTEM)
read Path_To_File   -> Read a file (eg: read /windows/win.ini)
get  Path_To_File   -> Download a file (eg: get users/administrator/desktop/password.txt)
help                -> Print this message.
exit                -> Exit this shell and return in relay mode.
                       If you want to quit type exit and then use CRTL-C

Any other command than that will be run as SYSTEM on the target.

C:\Windows\system32\:#whoami
[+] Creating service
[+] Service name: TK[redacted]m with display name: c[redacted]E successfully created, name r[redacted].bat
nt authority\system
```

Вероятность реализации подобной атаки высока. В крупных сетевых инфраструктурах распространена практика использования автоматических систем для инвентаризации ресурсов, установки обновлений, резервного копирования и других задач. Такие системы ежедневно подключаются к ресурсам домена и могут быть использованы нарушителями для атак.

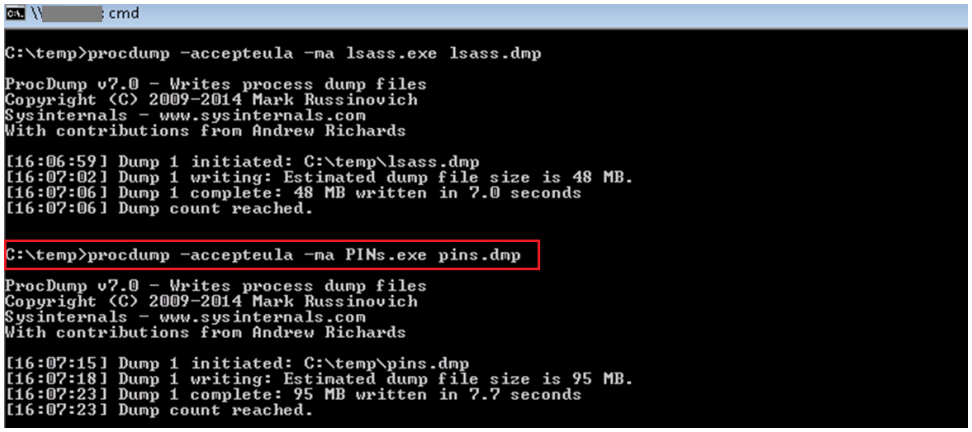
Рекомендации по защите. Для защиты от атаки необходимо реализовать подписывание SMB-пакетов (SMB Signing) на всех узлах сети, а также отключить протоколы NBNS и LLMNR. Кроме того, необходимо регулярно устанавливать актуальные обновления безопасности ОС.

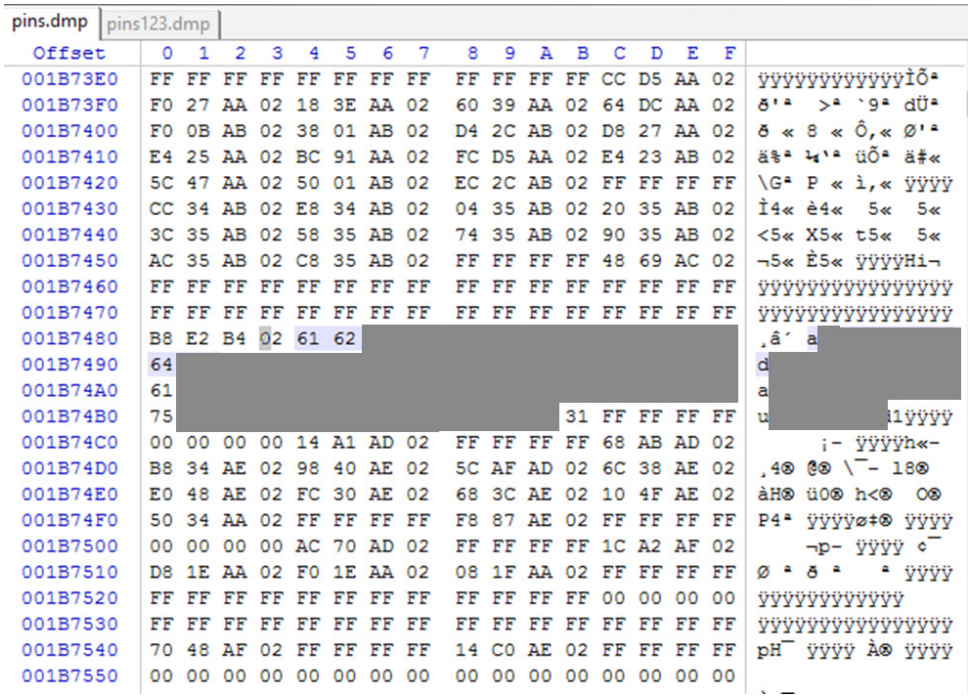
Сценарий 4. Чтение памяти процесса

Для развития атаки в ЛВС нарушитель может использовать имеющиеся у него привилегии, которые либо были получены в рамках первых шагов атаки (например, по сценариям 1, 2 или 3), либо изначально были у нарушителя (если он недобросовестный сотрудник компании). К примеру, нарушитель, обладающий привилегиями локального администратора на узле, может сохранить дамп памяти процессов ОС. Более того, такой уровень привилегий не всегда необходим, в общем случае достаточно привилегий того пользователя, от имени которого такие процессы запущены. Далее этот дамп может быть использован для получения чувствительной информации. В сценарии 1 приведен пример того, как может быть использован дамп процесса lsass, в данном же сценарии будет рассмотрено другое применение этой атаки.

Для безопасного хранения паролей многие администраторы используют специализированные утилиты. В данном сценарии продемонстрирована атака с получением ключа

доступа к файлам ПО PINs, в которых хранятся пароли для доступа ко множеству критически важных систем атакуемой организации. На рисунках ниже показано, как с помощью общедоступного ПО procdump был получен дамп памяти процесса PINs.exe, а в самом дампе найден пароль a**********1.





В результате атаки нарушитель получает возможность подключаться к критически важным системам с полученными паролями.

Рекомендации по защите. Для реализации атаки нарушителю необходим определенный уровень привилегий. Если процесс запущен от имени локального администратора, то ограничение привилегий пользователя ОС поможет защититься. Однако нарушитель сможет читать память тех процессов, которые запущены от имени такого пользователя (как показано в рассмотренном примере). Поэтому для защиты необходимо в первую очередь предотвратить несанкционированный доступ нарушителя к ОС, для чего обязательно должны применяться такие меры, как реализация строгой парольной политики, регулярное обновление ПО, защита от подбора учетных записей.

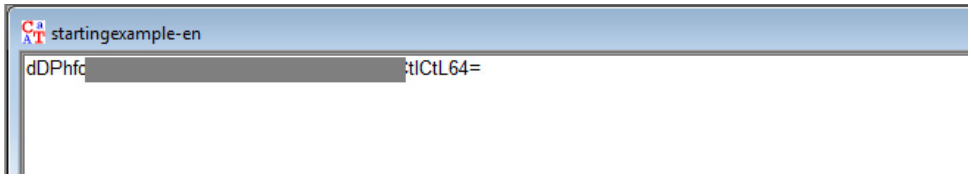
Сценарий 5. Групповые политики

Менее распространен в 2016 году, однако довольно часто встречался в предыдущие годы сценарий, основанный на использовании администраторами доменов групповых политик для смены паролей локальных администраторов. Зачастую привилегированные пользователи домена, создавая такие политики на контроллере домена (в директории \sysvol), не руководствуются принципами безопасности и вносят учетные данные в файл групповой политики. Для кодирования пароля применяется алгоритм AES, однако ключ шифрования является общедоступным и опубликован на сайте msdn.microsoft.com[15]. Таким образом, нарушитель, обладающий привилегиями пользователя домена, может получить учетные данные локальных администраторов на множестве узлов ЛВС.

```
<?xml version="1.0" encoding="utf-8"?>
<Groups clsid="{3125E937-EB16-4b4c-9934-544FC6D24D26}"><User
clsid="{DF5F1855-51E5-4d24-8B1A-D9BDE98BA1D1}" name="rezerv" image="2"
changed="2014-01-24 10:18:12" uid="{CBAE26BC-4205-49CF-BEE3-20ED0894376F}"
userContext="0" removePolicy="0"><Properties action="U" newName="" fullName=""
description="" cpassword="dDPhfo[redacted]CtICtL64"
changeLogon="0" noChange="1" neverExpires="1" acctDisabled="0" subAuthority=""
userName="rezerv"/></User>
 <User clsid="{DF5F1855-51E5-4d24-8B1A-D9BDE98BA1D1}" name="rezerv1" image="2"
userContext="0" removePolicy="0" changed="2014-01-24 12:20:05"
uid="{F6270CD4-B9BC-4D17-B775-F53A28CAA4B4}"><Properties action="U" newName=""
fullName="" description="" cpassword="x[redacted]zk
" changeLogon="0" noChange="1" neverExpires="1" acctDisabled="1" subAuthority=""
userName="rezerv1"/></User>
 <User clsid="{DF5F1855-51E5-4d24-8B1A-D9BDE98BA1D1}" name="prvd" image="3"
changed="2014-01-27 05:48:30" uid="{96BE9E27-E2F3-40FA-8D44-0BA0E9CC61AD}"
userContext="0" removePolicy="0"><Properties action="D" userName="prvd"/></User>
 <User clsid="{DF5F1855-51E5-4d24-8B1A-D9BDE98BA1D1}" name="Администратор
(встроенная учетная запись)" image="2" changed="2014-01-24 10:49:24"
uid="{5E74CA69-27C6-4C45-B729-70759C18B100}"><Properties action="U"
newName="Администратор" fullName="Администратор" description=""
cpassword="3Dh[redacted]Dk" changeLogon="0"
noChange="1" neverExpires="1" acctDisabled="1" subAuthority="RID_ADMIN"
userName="Администратор (встроенная учетная запись)"/></User>
 <Group clsid="{6D4A79E4-529C-4481-ABD0-F5BD7EA93BA7}" name="Администраторы
(встроенная учетная запись)" image="2" changed="2014-01-27 05:47:24"
uid="{525907BC-518C-47E1-BD9E-951538985D1D}" userContext="0"
removePolicy="0"><Properties action="U" newName="" description=""
deleteAllUsers="0" deleteAllGroups="0" removeAccounts="0" groupSid="S-1-5-32-544"
groupName="Администраторы (встроенная учетная запись)"><Members><Member
name="rezerv" action="ADD" sid=""/><Member name="rezerv1" action="ADD"
sid=""/><Member name="[redacted]" action="ADD"
sid="S-1-5-21-606747145-602609370-839522115-13304"/><Member
name="[redacted]" action="ADD" sid="S-1-5-21-606747145-602609370-839522115-1
8522"/><Member name="[redacted]Domain Admins" action="ADD"
sid="S-1-5-21-606747145-602609370-839522115-512"/><Member name="[redacted]Prvd"
action="ADD" sid="S-1-5-21-606747145-602609370-839522115-1569"/></Members></Prope
rties></Group>
</Groups>
```

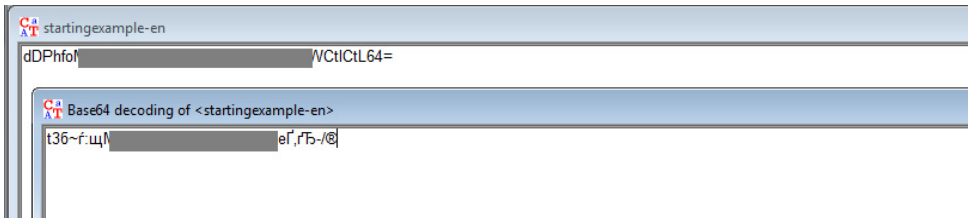
Расшифровка пароля может производиться следующим образом:

1. К зашифрованному паролю dDPhfo**********ICtL64 добавляются справа знаки равенства таким образом, чтобы длина полученной строки была кратна 4.



[15] https://msdn.microsoft.com/en-us/library/cc422924.aspx

2. Данная строка декодируется из base64-представления.



3. Полученная строка расшифровывается по алгоритму AES с помощью ключа, доступного по адресу msdn.microsoft.com/en-us/library/cc422924.aspx.

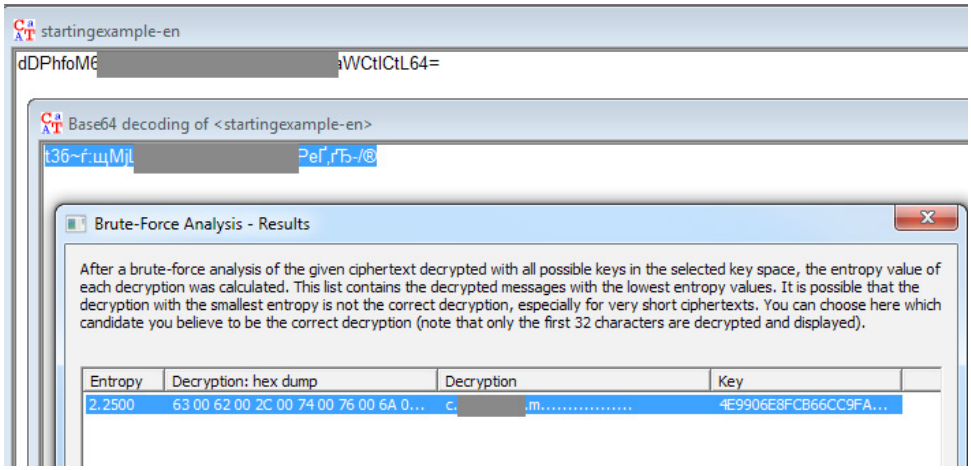
2.2.1.1.4 Password Encryption

7 out of 8 rated this helpful - Rate this topic

All passwords are encrypted using a derived Advanced Encryption Standard (AES) key.<3>

The 32-byte AES key is as follows:

```
4e 99 06 e8  fc b6 6c c9  fa f4 93 10  62 0f fe e8
f4 96 e8 06  cc 05 79 90  20 9b 09 a4  33 b6 6c 1b
```



4. Восстановлен пароль c******m.

Данный сценарий атаки требует наличия у атакующего доступа к файлам групповых политик. Сотрудник компании, являющийся пользователем домена, может обладать такими привилегиями, либо они могут быть получены по сценариям атак 1 и 2.

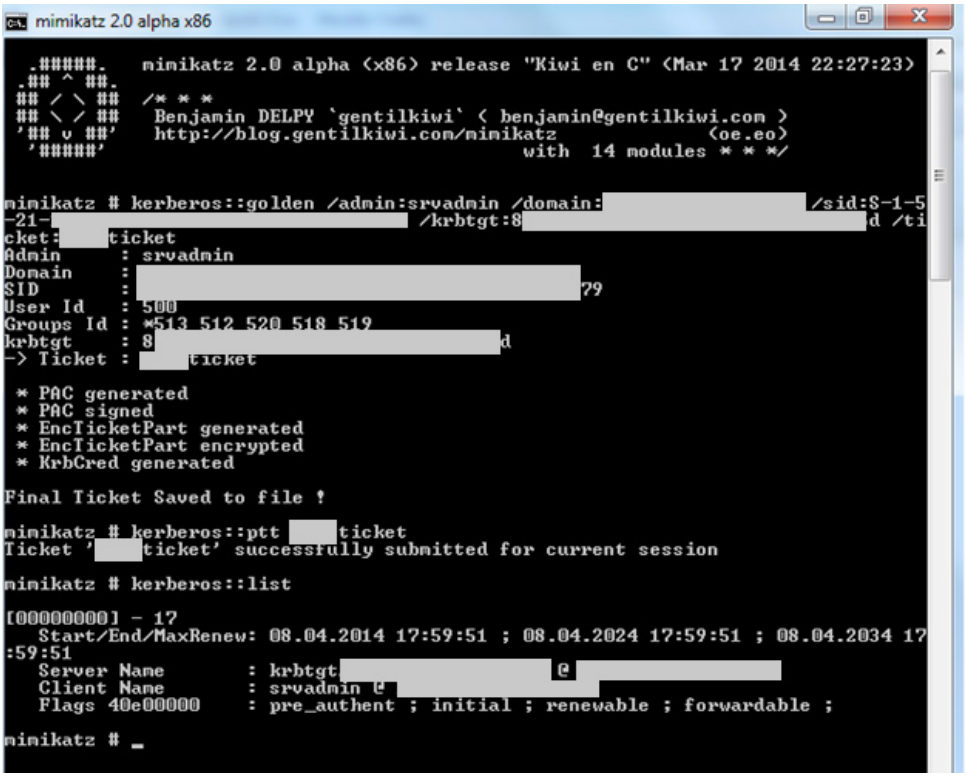
Рекомендации по защите. Подобный механизм изменения паролей локальных администраторов широко применяется в различных КИС, особенно он удобен для администрирования крупных распределенных инфраструктур. При таком подходе администратору не приходится подключаться к каждому из узлов, где необходима смена пароля, он получает возможность сделать это централизованно. Рекомендовать в данном случае можно либо полный отказ от использования такого подхода, либо создание таких политик только на ограниченное время, в которое происходит смена паролей, и удаление политик сразу же после выполнения операции. При этом необходимо принимать риски компрометации узлов сети.

Сценарий 6. Золотой билет Kerberos

Мы решили выделить эту атаку в отдельный сценарий в виду ее чрезвычайной опасности для КИС, хотя она требует первоначального получения соответствующего уровня привилегий. Атака основана на генерации билета доступа Kerberos пользователя на основе NTLM-хеша служебной учетной записи krbtgt и возможна из-за особенностей архитектуры протокола Kerberos и ОС семейства Windows.

Протокол Kerberos базируется на ticket-системе, то есть на предоставлении билетов доступа к ресурсам доменной инфраструктуры. Нарушитель способен создавать golden ticket на получение доступа любого уровня привилегий и может, соответственно, обращаться к ресурсам домена с максимальными привилегиями.

Атака возможна только при наличии у атакующего NTLM-хеша пароля krbtgt, получение которого, в свою очередь, возможно при наличии у атакующего актуальной резервной копии Active Directory либо привилегий в домене, позволяющих сделать такую копию (например, администратора домена). При этом в случае успешной атаки дальнейшее обнаружение действий злоумышленника, использующего аутентификацию по Kerberos, является крайне затруднительным, а смена паролей учетных записей, для которых были сгенерированы билеты доступа, не позволяет защититься.



Рекомендации по защите. В случае успешной компрометации системы развитие атаки может быть предотвращено только путем смены пароля[16] пользователя krbtgt, что сопряжено с перезапуском служб, использующих доменную аутентификацию. При этом стоит учитывать, что сама по себе смена пароля krbtgt не исключает возможности повторного получения атакующим NTLM-хеша пароля krbtgt, если у него сохранились первоначальные привилегии, используемые для атаки.

[16] http://technet.microsoft.com/en-us/library/8e3e4377-ef54-4a70-9215-a5d2ba4d0eb9(v=ws.10)#BKMK_Resetkrbtgt

Во избежание подобных атак рекомендуется обеспечить защиту привилегированных учетных записей (в частности, тех, что позволяют проводить резервное копирование Active Directory), в том числе с использованием средств двухфакторной аутентификации, а также обеспечить защиту резервных копий службы каталогов. Кроме того, важно защитить рабочие станции и серверы от атак с использованием утилит для получения учетных данных в открытом виде, в частности Mimikatz.

Сценарий 7. Pass the hash и pass the ticket. Атака на двухфакторную аутентификацию

В примере выше мы советовали использовать двухфакторную аутентификацию для защиты привилегированных учетных записей критически важных систем — например, контроллеров домена. Однако это не означает, что двухфакторная аутентификация сама по себе полностью защищает от атак. Скорее, это один из необходимых шагов при построении комплексной защиты КИС. Следующий сценарий демонстрирует уязвимости механизма двухфакторной аутентификации в ОС Windows.

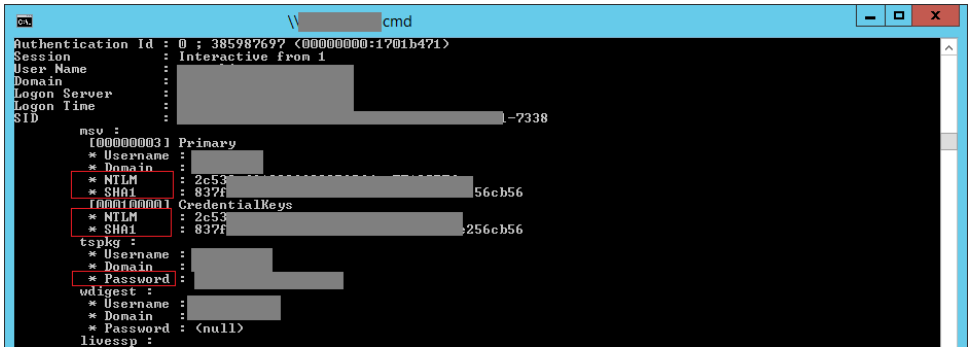
Аутентификация в ОС Windows возможна как по идентификатору и паролю, так и с использованием смарт-карты. Администратор может настроить систему так, чтобы она запрашивала исключительно смарт-карту для доступа к ОС либо предоставляла выбор метода аутентификации пользователю.

Принцип двухфакторной аутентификации подразумевает, что пользователь должен не только знать что-то (например, PIN-код или пароль), но и обладать чем-то (в данном случае — смарт-картой с установленным сертификатом). Только предъявив смарт-карту с корректным сертификатом и введя верный PIN-код, пользователь получает доступ к ОС.

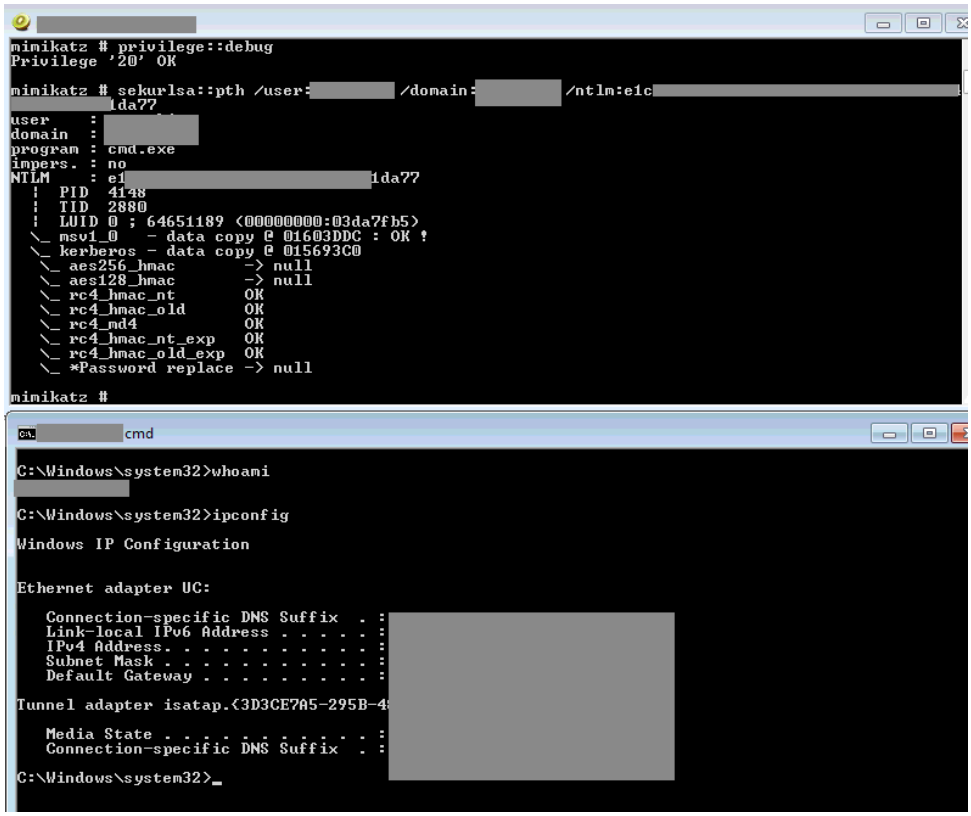
Когда в конфигурации учетной записи домена устанавливается атрибут, отвечающий за аутентификацию по смарт-карте, этой учетной записи присваивается некоторый NT-хеш. Его значение вычисляется случайным образом и неизменно при всех последующих подключениях к ресурсам домена. Контроллер домена отправляет этот хеш на узел, к которому подключается пользователь, при каждой аутентификации.

Уязвимость заключается в том, что злоумышленник может получить этот NT-хеш и использовать его при аутентификации методом pass the hash. Таким образом, злоумышленнику уже не нужно обладать смарт-картой и знать ее PIN-код, нарушается принцип двух факторов. А учитывая, что этот хеш постоянен, нарушитель получает возможность атаковать ресурсы домена с привилегиями скомпрометированной учетной записи на неограниченный период времени.

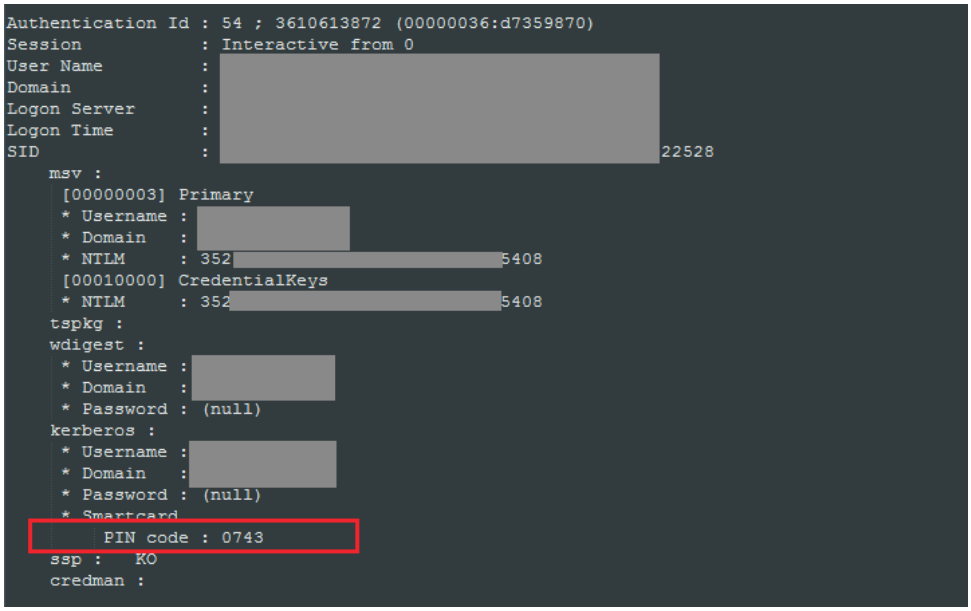
Для того чтобы получить NT-хеш, злоумышленник может использовать результаты запуска утилиты Mimikatz на узлах КИС в рамках атак по сценариям 1, 2 или 3.



На рисунке выше показан запуск Mimikatz на одном из узлов КИС, а на следующем рисунке продемонстрирован результат успешной аутентификации методом pass the hash с использованием полученного хеша пользователя. Этот пользователь входил в группу администраторов серверов, и для него была настроена аутентификация только по смарт-карте.



Кроме NT-хеша и пароля пользователя, злоумышленник может получить и PIN-код смарт-карты в открытом виде.



По сути, если злоумышленник может запускать утилиту Mimikatz на узлах КИС (непосредственно в ОС либо с использованием любого из возможных методов обхода средств защиты), он получает возможность компрометировать учетные записи привилегированных пользователей домена — даже при использовании двухфакторной аутентификации. Механизмы аутентификации в ОС Windows построены таким образом, что даже если нарушитель не сможет получить учетную запись администратора, он получит NT-хеш (генерируемый контроллером домена при использовании смарт-карты) либо билет Kerberos[17]. Если NT-хеш не изменяется и не имеет срока действия и может быть использован на любом узле КИС (в том числе на контроллере домена), то билет Kerberos выдается лишь на доступ к данному узлу на 10 часов и может быть продлен в течение недели. Оба эти значения могут быть использованы злоумышленником для аутентификации в обход двухфакторного механизма, для атак pass the hash и pass the ticket.

Рекомендации по защите. В Windows 10 реализована система Remote Credential Guard[18], которая призвана обеспечить защиту учетных записей при осуществлении удаленного доступа к ресурсам. В рамках тестирований на проникновение наши эксперты еще не встречались с использованием этой системы в КИС, а значит, исследование ее безопасности — дело ближайшего будущего. Судя по описанию производителя, использование Remote Credential Guard существенно повысит защищенность ресурсов КИС от атак методом pass the hash.

Заключение

Представленные в отчете сценарии атак — лишь часть техник, которые используются в работах по тестированию на проникновение. Некоторые атаки на КИС реализуются существенно сложнее, однако базируются на описанных здесь сценариях.

Данный отчет призван обратить внимание администраторов систем, сотрудников подразделений информационной безопасности и их руководителей на то, что атаки на ресурсы КИС вполне предсказуемы. Каждый из описанных сценариев основан на эксплуатации наиболее распространенных уязвимостей КИС, которые могут быть устранены путем изменения конфигурации либо с минимальными финансовыми вложениями. Кроме того, описания уязвимостей и недостатков защиты КИС ежегодно публикуются в аналитических исследованиях[19].

Необходимо также отметить, что сложность компрометации ресурсов в значительной степени зависит от того, является ли подход к защите комплексным. Даже в случае применения дорогостоящих решений по обеспечению безопасности они могут оказаться бесполезными, если пользователи и администраторы ресурсов применяют словарные пароли. В нашей практике было множество примеров, когда словарный пароль лишь одного пользователя позволял развить вектор атак в ЛВС до получения полного контроля над всей инфраструктурой корпоративной сети. Также было показано, что, получив привилегии локального администратора на рабочей станции или сервере, нарушитель может использовать специализированные утилиты для получения учетных данных даже при наличии антивируса.

Таким образом, организацию безопасной КИС необходимо начинать с базовых принципов:

+ использовать строгую парольную политику,
+ защищать привилегированные учетные записи,
+ повышать осведомленность сотрудников в вопросах ИБ,

[17] https://technet.microsoft.com/en-us/library/cc780469(v=ws.10).aspx
[18] https://technet.microsoft.com/ru-ru/itpro/windows/keep-secure/remote-credential-guard
[19] https://www.ptsecurity.com/ru-ru/research/analytics/

- не хранить чувствительную информацию в открытом виде,
- ограничить число интерфейсов сетевых служб, доступных на периметре,
- защищать либо отключать неиспользуемые протоколы канального или сетевого уровня,
- разделять сеть на сегменты, минимизировать привилегии пользователей и служб,
- регулярно обновлять ПО и устанавливать обновления безопасности ОС,
- регулярно проводить тестирование на проникновение КИС и анализ защищенности веб-приложений на периметре.

При этом важно обеспечить все эти меры в комплексе, только тогда защита будет эффективной, а затраты на дорогостоящие средства безопасности окажутся оправданы.

О компании

Positive Technologies — один из лидеров европейского рынка систем анализа защищенности и соответствия стандартам, а также защиты веб-приложений. Деятельность компании лицензирована Минобороны России, ФСБ России и ФСТЭК России, продукция сертифицирована ФСТЭК России и в системе добровольной сертификации «Газпромсерт». Организации во многих странах мира используют решения Positive Technologies для оценки уровня безопасности своих сетей и приложений, для выполнения требований регулирующих организаций и блокирования атак в режиме реального времени. Благодаря многолетним исследованиям специалисты Positive Technologies заслужили репутацию экспертов международного уровня в вопросах защиты SCADA- и ERP-систем, крупнейших банков и телеком-операторов.

POSITIVE TECHNOLOGIES

ptsecurity.com
pt@ptsecurity.com

facebook.com/PositiveTechnologies

facebook.com/PHDays