3. Чудненко Ю.В. Обзор: ИТ в органах государственной власти 2013 (Алексей Мошков: «Ни одно преступление в сфере ИТ не останется безнаказанным») // CNews: Аналитика. 2014. №187. С. 12.
4. Осипенко К.Ю. Жертвы и последствия киберпреступлений NORTON REPORT 2013 // Information Security/ Информационная безопасность. 2013. №5. С. 25.
5. Поляков В.В. Об использовании новых понятий при доказывании преступлений в сфере компьютерной информации // Российская юридическая наука: состояние, проблемы, перспективы: матер. Всеросс. науч.-практ. конф., посвященной 45-летию юридического образования на Алтае, 19-20 сентября 2008. – Барнаул: Изд-во АлтГУ, 2008. С 427 - 431.
6. Поляков В.В. Особенности расследования неправомерного удаленного доступа к компьютерной информации: дис. канд. юрид. наук: 12.00.09. Омск, 2008. 247 с.
7. Поляков В.В. Программное обеспечение, используемое для совершения компьютерных преступлений // Ломоносовские чтения на Алтае–2013: матер. Междунар. молодежной школы-семинара (Барнаул, 5-8 ноября 2013 г.). – Барнаул: Изд-во Алт. ун-та, 2013. Ч. 2 . С. 15-17.
8. Мамедов Н.А. Криминалистические проблемы расследования преступлений в сфере компьютерной информации // Юрист. 2008. №9. С. 32.

АУТЕНТИФИКАЦИЯ КАК СРЕДСТВО ОБЕСПЕЧЕНИИЯ БЕЗОПАСНОСТИ СЕРВЕРОВ НА ОСНОВЕ ClearOS

А.А. Егораев, АлтГТУ, факультет информационных технологий, 4к.
Научный руководитель – ***Е.В. Шарлаев***, к.т.н., доцент.

В условиях современной реальности уже трудно представить какую-нибудь крупную компанию без сосредоточенного информационного управляющего ресурса – сервера, так как с каждым днем человек все более и более стремиться автоматизировать весь процесс своей работы. В офисах появляются множество компьютеров, нормальное функционирование которых без сервера представить сложно – растут вычислительные мощности, объемы

информации растут еще быстрее и обходиться без файлового сервера или же просто интернет шлюза очень сложно и зачастую просто неудобно.

На серверах храниться множество информации, цену которой порой невозможно рассчитать, поэтому деятельность по обеспечению безопасности сервера выходит на первое место, ведь чем дороже на серверах информация, тем больше находиться желающих, которые хотят ею обладать. На сегодняшний день используются множество серверов, которые исполняют глобальное количество сервисов и процессов и далеко не секрет что практически каждый сервис имеет ту или иную уязвимость и найти ее дело времени. Уязвимости находят даже в тех сервисах, где их совсем и не ждешь. Без своевременного реагирования и устранения проблем, являющихся следствием ошибочно порождаемого программного продукта, о безопасности системы можно забыть.

Сложно представить какие страшные последствия понесет компания, если на ресурсы ее сервера проникнет злоумышленник. К наиболее вероятностным способам для проникновения злоумышленника на вычислительные ресурсы относиться выполнение задач, которые являются следствием не декларированных возможностей различного ПО, присутствия программных закладок или результатом действия вирусного программного обеспечения. Одним из возможных решений в обеспечении безопасности сервера является применение собственных менеджеров процессов, которые позволяют разграничить их между пользователями, контролировать действия, назначать привилегии и права. Однако использование менеджера процессов без связки с двухфакторной аутентификацией [3] лишь немного затруднит и замедлит злоумышленника ведь количество эксплоитов и «дыр» растет и быть уверенным в том, что злоумышленник не сможет получить права суперпользователя, используя ту или иную «дыру» нельзя.

На сегодняшний день двухфакторная аутентификация используется довольно часто в WEB сфере, но при удаленном доступе к данным сервера этот механизм защиты до сих пор является редкостью и, как правило, ограничивается лишь использованием электронных замков и Е-токенов.

Целью настоящего исследования является поиск наиболее безопасного соединения с удаленным сервером. В качестве объекта исследования выбрана вычислительная сеть на основе сервер на

базе ClearOS с внешним подключением по SSH протоколу[1], т.е. использующее в качестве подключения клиента к серверу 22-ой программный порт.

Для начала необходимо сделать так, чтобы сервер при попытке подключения к нему клиента выглядел как обычный сервер, это поможет завести злоумышленника в заблуждение. Так же необходимо настроить подключение по SSH на контроль количества попыток ввода пароля, предпочтительно три раза, после чего отбрасываются все попытки соединения с сервером, при этом необходимость занести ip-адресс устройства клиента, который пытался инициировать соединение в так называемый «бан лист» (черный список) на один час, что обезопасит от использования «брут перебора» неопытных атакующих и замедлит опытных, так как вызовет у них необходимость в использовании дополнительных промежуточных узлов составной сети, например прокси-серверов, для автоматизации и анализа каким образом сервер-цель сбрасывает соединение. Для усиления защитных механизмов также необходимо установить ограничение на количества одновременно подключенных клиентов. После применения данных настроек можно оставить конфигурирование SSH-параметров и перейти ко второй стадии защиты.

Вторым фактором усиленной аутоинтефикации[2] является применение генерации случайных символов, а именно 6 символов. Программа реализована на языке C# с применением Mono. Это позволило реализовать универсальное приложение, которое можно использовать как на Unix платформах, так и на других. Приложение основано на клиент-серверной технологии. Серверная часть располагается непосредственно на самом сервере и работает как сервис, при ее запуске в конфигурационный файл записываются все запущенные приложения и закрывается к ним доступ ровно до того момента как сервис не пропустит к ним. Это позволяет оградить злоумышленника от интересующих файлов и ввести в легкое недоумение - ведь только что у него появилось окно приветствия подключения по SSH, а вслед за ним окно с 6 символами и окном ввода информации.

Рассмотрим работу серверной части поподробнее. Когда клиент подключается к серверу и проходит аутентификацию SSH, происходит генерация случайных символов, с комбинацией различных алгоритмов. Желательно использование не одного алго-

ритма генерации, а нескольких, в зависимости от времени или дня недели, например. День недели на мой взгляд намного лучше нежели время, так как его можно использовать как еще один символ, то есть у нас высвечивается на сервере окно с 6 символами плюс один скрытый который мы будем вводить после третьего видимого символа на нашем клиенте, при генерации кода на сервере это пройдет автоматически.

Обратим внимание на использование клиентского модуля. Клиентская часть используется как на Unix, так на Windows, но есть необходимость иногда подключиться к серверу моментально, используя, например, смартфон, поэтому клиентская часть адаптируется под андроид. Алгоритм работы клиентской части очень прост. Запускаем клиентскую часть, видим окно ввода, вводим в поля символы, которые представил нам сервер и не забываем, что после третьего символа должен идти символ, отвечающий за день недели, после чего клиентская часть выдает нам с генерированный по этим данным ключ для подключения. Вводим его в окно на сервере и получаем после этого доступ ко всем функциям системы. Подобрать такой с генерированный пароль довольно сложно, но все же лучше сделать следующее: ограничить количество попыток ввода двумя, после второй, неправильной попытки, будет происходить разрыв соединения с сервером и клиенту придется заново осуществлять подключение и аутентификацию по SSH, при этом вновь произведётся генерация пароля уже нового, что сделает перебор невозможным

Данный способ прохождения аутентификации не делает сервер полностью безопасным, но он усложняет действия злоумышленников. А так как используется помимо двухфакторной аутентификации еще и менеджер процессов, то злоумышленнику придется потратить много времени и сил, чтобы проникнуть и получить нужные ему файлы.

Список литературы

1. Настройка сервера SSH (теория и практика). [Электронный ресурс]. – Режим доступа: http://www.nixp.ru/articles/Настройка-сервера-SSH-(теория-и-практика).html
2. Контроль доступа (информатика) / Материал из Википедии – свободной энциклопедии, [Электронный ресурс]. – Режим до-

ступа: http://ru.wikipedia.org/wiki/Контроль_доступа (информатика)/
3. Что такое двухфакторная аутентификация. Юрий Медяков 14.01.2014 / [Электронный ресурс]. – Режим доступа: https://cryptostore.ru/article/obzory/chto_takoe_dvukhfaktornaya_autentifikatsiya/

ПРИМЕНЕНИЕ DATA LOSS PREVENTION ТЕХНОЛОГИЙ ДЛЯ ЗАЩИТЫ ОТ ИНСАЙДЕРСКИХ УГРОЗ

И.А. Красников, АлтГТУ, факультет информационных технологий, 5 к.
Научный руководитель – ***Е.В. Шарлаев***, к.т.н., доцент.

Распространение информационных технологий и тенденция подключения государственных информационных систем (ГИС) к информационно-телекомуникационным сетям общего пользования являются порождающим фактором увеличения риска реализации инсайдерских угроз информационной безопасности организаций. Согласно нормативно-правовым актам (НПА) в области обеспечения защиты информации в ГИС, таких как ФЗ №152 «О персональных данных», Постановление правительства 1119 «Об утверждении требований к защите персональных данных при их обработке в информационных системах персональных данных», Указ Президента РФ от 17.03.2008 N 351 "О мерах по обеспечению информационной безопасности Российской Федерации при использовании информационно-телекоммуникационных сетей международного информационного обмена", Приказ ФТЭК от 13.02.2013 №17 «Об утверждении требований о защите информации, не составляющей государственную тайну, содержащейся в государственных информационных системах», Приказ ФСТЭК 18.02.13 №21 «Об утверждении состава и содержания организационных и технических мер по обеспечению безопасности персональных данных при их обработке в информационных системах персональных данных» и Руководящие документы ФСТЭК, необходимо использование сертифицированных по требованиям безопасности средств защиты информации. Делая акцент на увеличение количества утечек конфиденциальной информации в сеть, следует предпринимать своевременные меры по нейтрализации