DOI 10.37882/2223–2966.2021.12–2.09

РАЗРАБОТКА МЕТОДИКИ ОРГАНИЗАЦИИ УДАЛЕННОГО УПРАВЛЕНИЯ АРМ В СОЕДИНЕНИИ

DEVELOPMENT OF A METHOD FOR ORGANIZATION OF REMOTE CONTROL OF AWPS IN A CONNECTION

A. Pavlov
A. Gladkov

The article discusses the features of developing a methodology for organizing remote control of AWPs in a connection. Special attention is paid to the description of the purpose and distinctive features of using data transfer protocols such as RDP and SSH. In addition, the capabilities of the designated protocols, their advantages, disadvantages, and vulnerabilities are indicated and described in detail. The method of organizing remote control is described using the example of an automated workstation in the formation of the National Guard of the Russian Federation. Using this technique in practice will improve the security of the network connection and the transmission of user data.

Keywords: control, access, protocols, security, vulnerability, protection, user, network.

Павлов Александр Николаевич
К.т.н., доцент, Пермский военный институт войск национальной гвардии Российской Федерации
pal0707@mail.ru
Гладков Алексей Николаевич
К.т.н., доцент, Пермский военный институт войск национальной гвардии Российской Федерации
vmkcic@yandex.ru

В статье рассмотрены особенности разработки методики организации удаленного управления АРМ в соединении. Отдельное внимание уделено описанию назначения и отличительным чертам использования протоколов передачи данных, таких как RDP и SSH. Кроме того, обозначены и детально описаны возможности обозначенных протоколов, их достоинства, недостатки, а также уязвимости. На примере АРМ в соединении Войск Национальной Гвардии Российской Федерации описана методика организации удалённого управления. Использование данной методики на практике позволит повысить безопасность сетевого соединения и передачи данных пользователей.

Ключевые слова: управление, доступ, протоколы, безопасность, уязвимость, защита, пользователь, сеть.

Введение

Осуществление удаленного управления автоматизированными рабочими местами пользователей является значимым направлением в сфере автоматизации. Благодаря такому способу организации управления, эффективность выполняемых задач специалистами информационных технологий возрастает в разы. Однако, в связи с переходом войск Национальной гвардии на продукты отечественного производства, появилась необходимость переподготовки и обучения данных специалистов для администрирования рабочих станций, использующих ОС «AstraLinux». Таким образом, поставлена задача создания методики организации удаленного управления АРМ.

Удаленное подключение ко всем АРМ в соединении позволяет при необходимости выявить проблему и принять меры для ее устранения стационарно, без дополнительных передвижений и временных затрат.

Важно отметить, что использование удаленного доступа к АРМ пользователей не снизит уровень информационной защищенности воинской части. Это обусловлено тем, что при удаленном подключении имеется возможность использовать протоколы с функцией шифрования данных и туннелирования каналов для их передачи, что минимизирует риски получения несанкционированного доступа к данным или возможностям ПО АРМ извне. Также не маловажным является факт того, что удаленный доступ, как правило, используется в защищенных и засекреченных каналах передачи данных в воинской части, а значит, в данной локально-вычислительной сети имеются криптомаршрутизаторы, что также позволяет поддерживать безопасность соединения.

Назначение и использование протоколов передачи данных

Для осуществления удаленного подключения используются протоколы RDP и SSH, однако они имеют существенные различия. Рассмотрим оба протокола и их особенности.

Протокол RDP

RemoteDesktop используется для подключения к удаленному компьютеру или виртуальным приложениям и рабочим столам, предоставленных администра-

тором. Протокол RDP основан на семействе стандартов протокола T-120 и является его расширением. Многоканальный протокол позволяет использовать отдельные виртуальные каналы для передачи разнообразной информации Приложение позволяет продуктивно работать независимо от того, где находится пользователь. Клиенты Microsoft Remote Desktop могут использовать и контролировать удаленный ПК. С помощью Remote Desktop есть возможность делать с удаленным ПК все то же самое, что и с физическим ПК, например:

1. Использовать приложения, установленные на удаленном ПК.
2. Доступ к файлам и сетевым ресурсам на удаленном ПК.
3. Оставлять приложения открытыми после выключения клиента.

RDP является расширением основного протокола T. Share. Некоторые другие возможности сохраняются как часть RDP, например, архитектурные особенности, необходимые для поддержки многоточечных (многосторонних сеансов). Многоточечная доставка данных позволяет доставлять данные из приложения в режиме реального времени нескольким сторонам, например, виртуальным доскам. При этом не требуется отправлять одни и те же данные в каждый сеанс по отдельности.

Безопасность при использовании RDP

Сеансы удаленного рабочего стола работают по зашифрованному каналу, не позволяя никому просматривать сеанс пользователя, прослушивая сеть. Однако в методе, используемом для шифрования сеансов в ранних версиях RDP, существует уязвимость. Эта уязвимость может позволить несанкционированный доступ к сессии с помощью атаки, которая получила название «человек внутри».

В идеальных и контролируемых средах протокол удаленного рабочего стола работает отлично. Однако защита RDP для предотвращения несанкционированных сеансов, перехвата, неправомерного доступа, эксплойтов, повышения привилегий и т.д. требует уровня зрелости ИТ-безопасности, который выходит далеко за рамки стандартных настроек RDP.

Настройки RDP по умолчанию обеспечивают только базовый уровень шифрования и базовую безопасность. Если полагаться только на эти параметры безопасности они создают ситуацию, представляющую неприемлемый риск для большинства организаций.

Первое правило безопасности RDP — никогда не оставлять RDP открытым для доступа в Интернет — независимо от того, насколько сильно укреплены конечные точки и системы. Риски такого доступа слишком высоки. RDP предназначен для использования только в локальной сети (LAN).

Также можно выделить несколько базовых приемов и средств защиты RDP:

- патчи: поддержка серверов в актуальном состоянии;
- использование сложных паролей, целесообразно также применять двухфакторную аутентификация и активно внедрять политику блокировки.
- изменение порта по умолчанию, вместо используемого RDP — 3389 следует выбрать другой через реестр;
- брандмауэр Windows: для защиты следует использовать встроенный брандмауэр Windows для ограничения сеансов RDP по IP-адресу.
- аутентификация на сетевом уровне (NLA): необходимо включить NLA, которая не используется по умолчанию в старых версиях;
- ограничение доступа RDP: доступ RDP для определенной группы пользователей должен быть ограничен, в первую очередь это должны быть администраторы домена;
- туннелирование доступа к RDP, туннелирование может осуществлять через IPSec или Secure Shell (SSH).

Протокол SSH

SSH, сокращение от Secure Shell,— это протокол удаленного администрирования и сетевой протокол, изначально разработанный для замены Telnet и других незащищенных протоколов, таких как Berkeley Remote Shell (rsh). SSH обеспечивает безопасное удаленное администрирование системы и передачу файлов по незащищенным сетям. Протокол SSH используется для защиты сетевых сервисов в незащищенной сети. В настоящее время он применяется практически во всех дата-центрах. Secure Shell использует архитектуру клиент-сервер для обеспечения безопасного канала в незащищенной сети. Особенности туннелирования SSH сессии представлены на рис. 1

Возможности SSH

Этот криптографический сетевой протокол нашел свое широкое применение для защиты всех видов сетевых услуг. Он использует шифрование для защиты соединения между SSH-клиентом и сервером, защищая от атак в сети. SSH позволяет получать доступ к командным строкам, выполнять команды, входить в систему и выполнять задачи системного администратора удаленно и безопасно Кроме того, протокол SSH

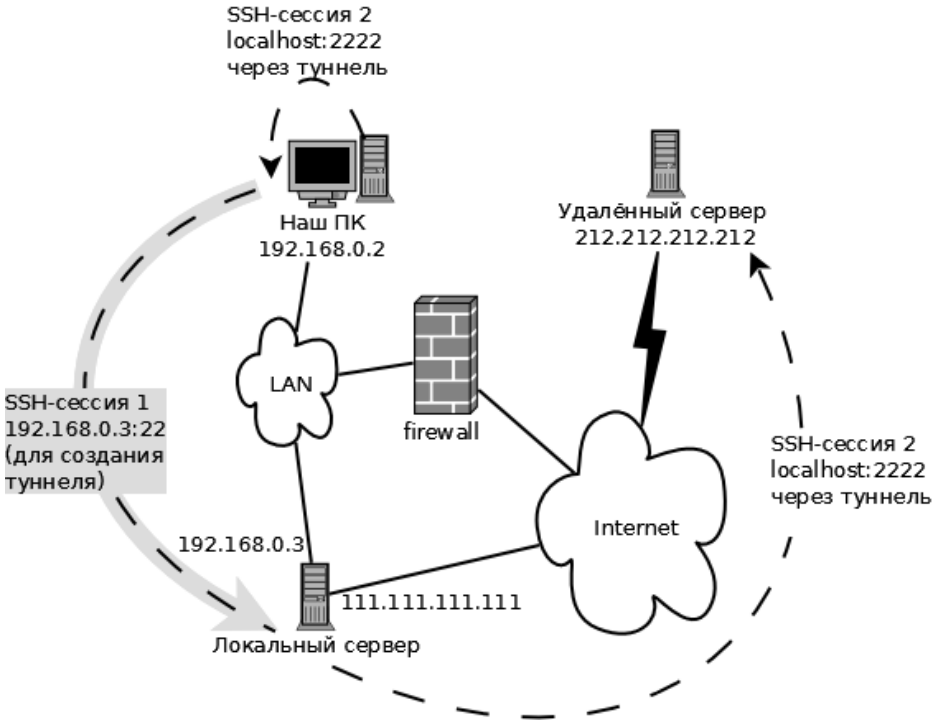

Рис. 1. Осуществление туннелирования канала в SSH сессии

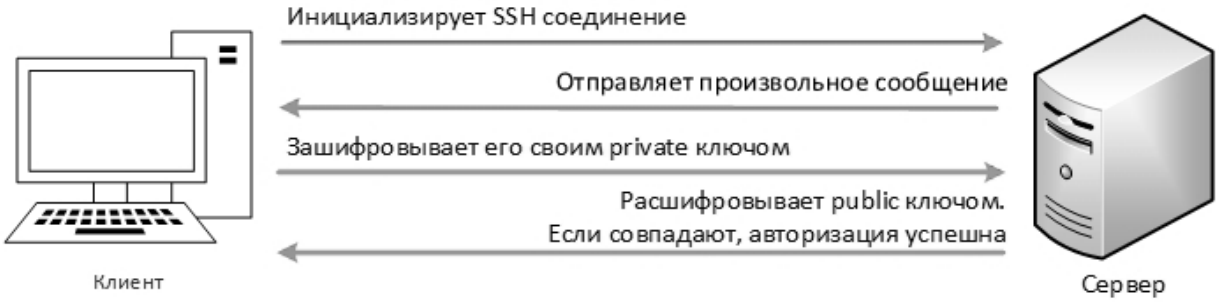

Рис. 2. Аутентификация по открытому ключу пользователя

также используется в различных механизмах передачи файлов.

Например:

1. SFTP (протокол передачи файлов SSH; безопасная альтернатива FTP).
2. FASP (быстрый и безопасный протокол).
3. SCP (Защищенная копия).

Шифрование протокола направлено на обеспечение строгой конфиденциальности и целостности данных. SSH использует криптографию с открытым ключом для своего механизма аутентификации, известного как «аутентификация с открытым ключом». Кроме того, он также поддерживает аутентификацию на основе пароля.

Основные способы использования SSH

1. Использование автоматически сгенерированных пар открытого и закрытого ключей для шифрования сетевого подключения и входа в систему с использованием пароля.
2. Использование сгенерированных вручную пар открытого и закрытого ключей для аутентификации. Таким образом пользователи и программы могут входить в систему без пароля.

Неизвестные открытые ключи должны всегда проверяться во всех версиях SSH, прежде чем они будут признаны действительными; во избежание авторизации неавторизованного злоумышленника как действительного пользователя (см. рис. 2).

В облачных вычислениях SSH полезен для решения проблем с подключением и предотвращения проблем с безопасностью. Туннель SSH может обеспечить безопасный путь через Интернет через брандмауэр, избегая доступа виртуальных машин непосредственно в Интернет.

Протокол SSH обеспечивает следующие меры безопасности:

1. После первоначального подключения клиент может проверить, что он подключается к тому же серверу, к которому он подключался ранее.
2. Клиент передает свою аутентификационную информацию на сервер, используя надежное 128-битное шифрование.
3. Все данные, отправленные и полученные во время сеанса, передаются с использованием 128-битного шифрования, что делает перехваченные передачи чрезвычайно сложными для дешифрования и чтения.
4. Клиент может пересылать приложения X11 с сервера. Этот метод, называемый пересылкой X11, обеспечивает безопасные средства использования графических приложений в сети.

Поскольку протокол SSH шифрует все, что он отправляет и получает, его можно использовать для защиты небезопасных протоколов.

Уязвимости протоколов

Протокол RDP

Наиболее важные уязвимости в RDP это:

1. С момента появления RDP различные версии имели множество уязвимостей, включая несколько, таких как BlueKeep и DejaBlue, которые позволяли удаленное выполнение кода и повышение привилегий. Для любой среды, использующей RDP, администраторам информационных технологий необходимо следить за обновлениями безопасности и своевременно их применять. Без многих из этих исправлений безопасности лишь немногие средства защиты смогут предотвратить эксплуатацию.
2. Неограниченный доступ к портам. RDP соединения почти всегда происходят через порт 3389*. Злоумышленники могут предположить, что это используемый порт, и применить его, в частности, для проведения атак по пути.

3. Лицензирование: Microsoft требует лицензирования протокола RDP для его использования в среде. Развертывание решений сторонних производителей или версий с открытым исходным кодом может нарушить лицензионные соглашения с Microsoft. Необходимо убедиться, что все сторонние решения, использующие RDP, которые развернуты, имеют соответствующую лицензию Microsoft.

Протокол SSH

Можно отметить четыре уязвимости SSH, которые нельзя игнорировать:

Проблемы с отслеживанием ключей SSH. Нередко типичное крупное предприятие с 10 000+ серверами имеет более миллиона ключей SSH, что делает поиск и управление каждым ключом невероятно трудным, если не невозможным. Организации обычно накапливают большое количество ключей SSH, потому что конечные пользователи могут создавать новые ключи SSH (учетные данные) или даже дублировать их без надзора, в отличие от сертификатов или паролей.

Совместное использование ключей. Для повышения эффективности ключи SSH часто используются совместно или реплицируются для общей группы сотрудников или серверов и компонентов инфраструктуры. В результате дублирования ключей SSH всего пять из 20 уникальных ключей могут предоставлять доступ ко всем машинам. Такой подход может облегчить работу ИТ-команд в краткосрочной перспективе, но он также облегчает жизнь злоумышленникам в долгосрочной перспективе. Дублирование ключей SSH приводит к созданию сложных связок между многими закрытыми и открытыми ключами, что значительно снижает уровень безопасности, поскольку трудно повернуть и отозвать один ключ, не нарушив при этом бесчисленное множество других связей между ключами SSH, которые используют один и тот же отпечаток ключа.

Статические SSH-ключи. Очевидным является тот факт, что ротация одного миллиона с лишним ключей SSH может стать логистическим кошмаром. Многие ИТ-администраторы и специалисты по безопасности редко меняют и перераспределяют ключи, опасаясь, что может быть забыт какой-либо критически важный компонент. Эти факторы обычно приводят к резкому увеличению количества статических ключей SSH, открывая злоумышленникам возможность скомпрометировать неизмененный ключ, использовать его для перемещения по сети и получить постоянный несанк-

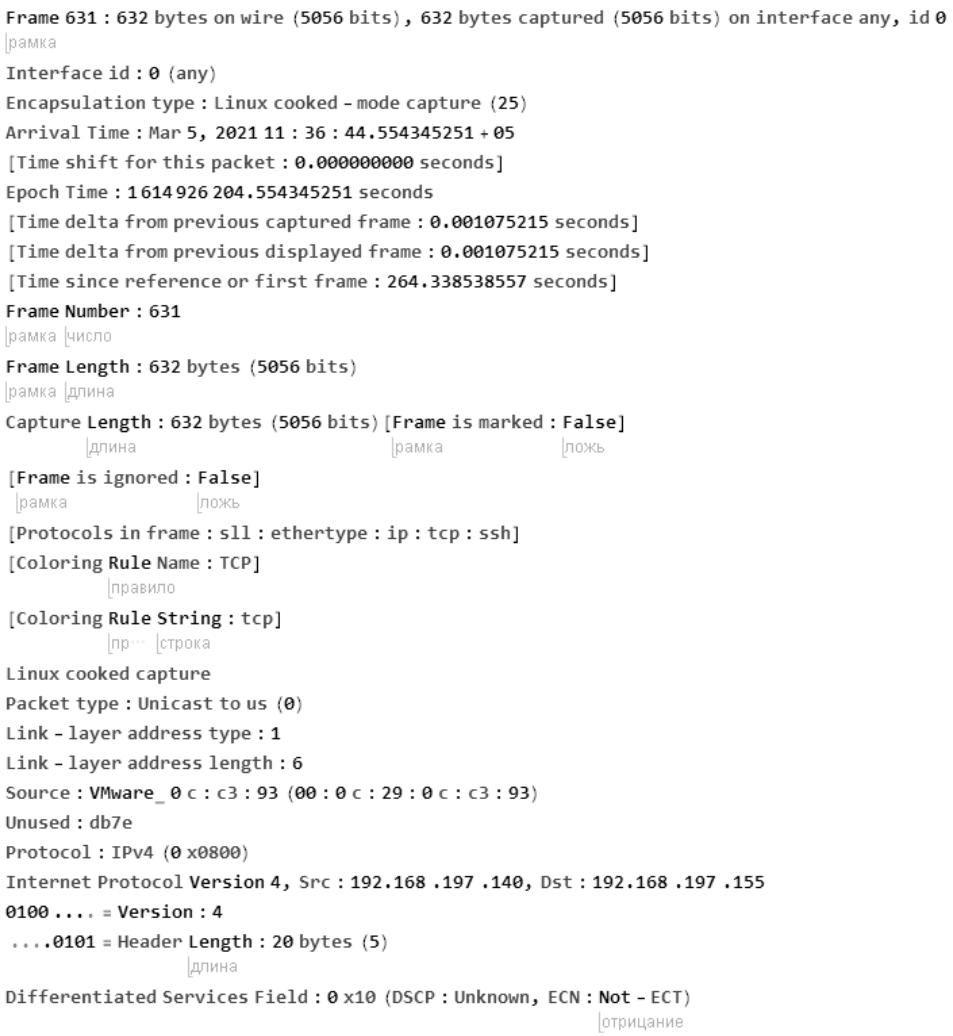

Рис. 3. Пример пакета Wireshark

ционированный доступ к конфиденциальным данным и активам.

Встроенные SSH-ключи — те, с которыми никто не хочет связываться. Ключи SSH часто встраиваются в приложения или сценарии. Администраторы часто боятся менять их, поскольку не понимают код, в который встроены ключи, или настоятельно не рекомендуют менять их из-за уровня координации, необходимого для предотвращения сбоев в работе системы. В результате статические SSH-ключи, встроенные в приложения, код и скрипты, могут привести к появлению постоянных «черных ходов» для злоумышленников.

При удаленном подключении с сервера к пользователю по протоколу SSH, мной был создан файл на компьютере пользователя и размещен там короткий текст. Пакеты, передаваемые при удаленном подключении, были наглядно получены при помощи использования программы Wireshark. Далее приведен пример этих пакетов (рис. 3).

Для дополнительного шифрования данных были использованы два алгоритма, представленных ранее — AESиRSA.

Далее представлен пример наложения шифра на пакеты, передаваемые при удаленном доступе.

Первым был использован алгоритм AES.

Для начала, необходимо создать симметричный ключ, который и будет накладываться на наши данные (рис. 4, 5).

Далее, созданный ключ применен к информации, после чего получен приведенный ниже код. Так будет выглядеть текст с наложенным на него шифром.

```
SymmetricKey_[Association["Cipher" → "AES256", "BlockMode" → "CBC",
  "Key" → ByteArray[{229, 200, 197, 114, 99, 205, 99, 168, 183, 86, 45, 198, 170,
     114, 57, 49, 191, 210, 173, 76, 128, 165, 3, 190, 109, 194, 10, 19, 79, 1, 4, 50}],
  "InitializationVector" → None]]
```

Рис. 4. Генерация симметричного ключа

```
Total Length : 616
сум··· длина
Identification : 0 xd838 (55 352)
Flags : 0 x4000, Don ' t fragment
Fragment offset : 0
Time to live : 64
Protocol : TCP (6)
Header checksum : 0 x53ce [validation disabled]
[Header checksum status : Unverified]
Source : 192.168 .197 .140
Destination : 192.168 .197 .155
Transmission Control Protocol, Src Port : 22, Dst Port : 35 692, Seq : 8438, Ack : 6575, Len : 564
            элемент управления
Source Port : 22
Destination Port : 35 692
[Stream index : 1]
[TCP Segment Len : 564]
Sequence number : 8438 (relative sequence number)
последовательность аргументов
Sequence number (raw) : 3 342 194 105
последовательность аргументов
[Next sequence number : 9002 (relative sequence number)]
Acknowledgment number : 6575 (relative ack number)
Acknowledgment number (raw) : 1 102 121 395
1000 .... = Header Length : 32 bytes (8)
            длина
Flags : 0 x018 (PSH, ACK)
```

Рис. 5. Пример пакета Indefication

```
( → 24,) → 24,_ → 1,- → 6,+ → 1,= → 3,[ → 20,] → 20,. → 30,, → 16,' → 1,
/ → 1,: → 76,\n → 60,→ 470,0 → 66,
1 → 47,2 → 36,3 → 29,4 → 25,5 → 46,6 → 30,7 → 10,8 → 18,9 → 18,a → 76,
A → 6,b → 23,c → 58,C → 16,d → 46,
D → 7,e → 174,E → 5,f → 22,F → 11,g → 19,h → 15,H → 5,i → 80,I → 4,
k → 16,K → 2,l → 40,L → 11,m → 45,M → 2,
n → 87,N → 9,o → 76,O → 5,p → 25,P → 19,q → 6,Q → 1,r → 90,R → 2,
s → 80,S → 16,t → 94,T → 16,u → 40,U → 6,
"v" → 14, "V" → 3, "w" → 11, "W" → 2, "x" → 11, "y" → 19, "z" → 3
```

Рис. 6

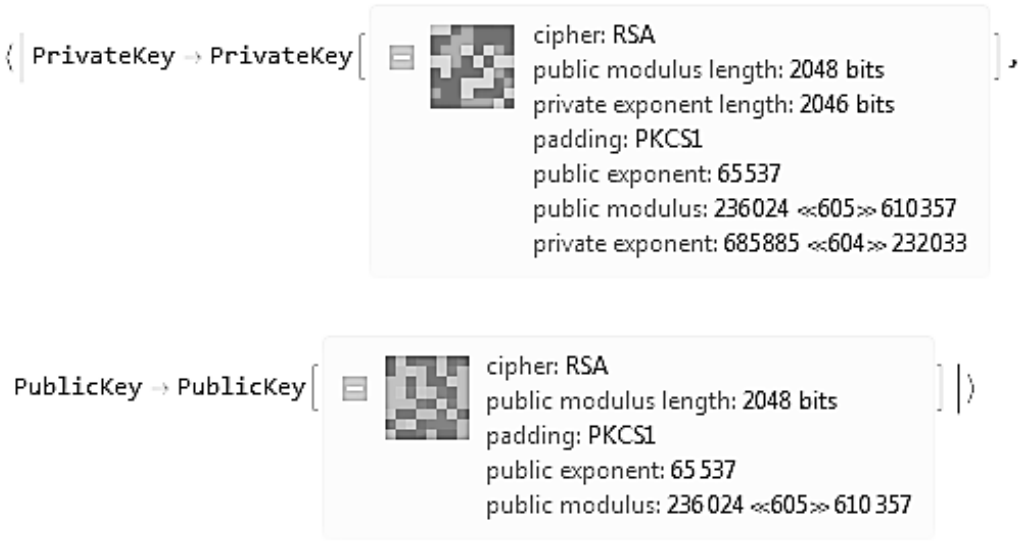

Рис. 7. Генерация связки ключей

В данном тексте представлено количество символов и букв в полученных ранее и зашифрованных пакетах (рис. 6).

Следующим алгоритмом шифрования был применен RSA.

Сначала, также необходимо создать ключи, но в данном случае создаются открытый и закрытый ключ. При создании данных ключей администратором, открытый ключ выдается пользователю, а закрытый остается у администратора и загружается на сервер, при шифровании данных, будет применяться закрытый ключ, пользователь же будет использовать открытый ключ, для проверки оригинальности файлов (рис. 7).

Заключение

Таким образом, для обучения специалистов информационных технологий и повышения их продуктивности предложена и разработана методика эксплуатации отечественного программного обеспечения. В данном случае — методика организации удаленного управления автоматизированным рабочим местом. При этом установка возможности удаленного администрирования повысит безопасность сетевого соединения и передачи данных пользователей. Также возрастет эффективность работы специалистов ИТ и увеличится их возможность контроля соблюдения необходимых требований пользователями.

ЛИТЕРАТУРА

1. Трушин В.А., Рева И.Л., Иванов А.В. Усовершенствование методики оценки разборчивости речи в задачах защиты информации // Ползуновский вестник № 3/2 2012. — С. 238–241.
2. Меньшаков Ю.К. Защита объектов и информации от технических средств разведки. — М.: Изд-во РГГУ, 2002.
3. Хорев А.А. Защита информации от утечки по техническим каналам. Технические каналы утечки информации. — М.: Гостехкомиссия РФ, 1998.
4. Халяпин Д.Б. Защита информации. — М.: Изд. дом «Мир безопасности», 2001.
5. Торокин А.А. Основы инженерно-технической защиты информации. — М.: Ось-89, 1998.
6. Чипига, А.Ф. Информационная безопасность автоматизированных систем / — М.: Гелиос АРВ, 2017. — 336 с.
7. Щеглов А.Ю. Защита компьютерной информации от несанкционированного доступа. — С.-П., 2004–384 с.
8. Шаньгин, В.Ф. Информационная безопасность и защита информации / В.Ф. Шаньгин. — М.: ДМК, 2014. — 702 с.
9. Запечников, С.В. Информационная безопасность открытых систем. В 2-х т. Т. 1 — Угрозы, уязвимости, атаки и подходы к защите / С.В. Запечников, Н.Г Милославская. — М.: ГЛТ, 2018. — 536 с.
10. Дейтел, Х.М. Операционные системы. Т. 2. Распределенные системы, сети, безопасность / Х.М. Дейтел, П.Д. Дейтел, Д.Р. Чофнес; Пер. с англ. С.М. Молявко, — М.: БИНОМ, 2013. — 704 с.

Журнал «Современная наука: актуальные проблемы теории и практики»