Министерство образования и науки Российской Федерации
НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Е.А. БАСЫНЯ

СИСТЕМНОЕ АДМИНИСТРИРОВАНИЕ И ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

Утверждено
Редакционно-издательским советом университета
в качестве учебного пособия

НОВОСИБИРСК
2018

УДК 004.056:004.383.2(075.8)
Б 278

Рецензенты:

д-р техн. наук, профессор кафедры автоматики *Г.А. Французова*

канд. техн. наук, науч. сотр. лаборатории индустриальной информатики ФГБУН ИВТ СО РАН *А.В. Сафронов*

Работа подготовлена на кафедре автоматики для студентов дневной формы обучения, проходящих подготовку по направлениям 27.03.04 «Управление в технических системах» и 09.03.01 «Информатика и вычислительная техника»

Басыня Е.А.

Б 278 Системное администрирование и информационная безопасность: учебное пособие / Е.А. Басыня. – Новосибирск: Изд-во НГТУ, 2018. – 79 с.

ISBN 978-5-7782-3484-0

В работе изложен анализ уязвимостей стека протоколов *TCP/IP*, операционных систем и программного обеспечения. Рассмотрены алгоритмы, методы, инструменты и средства их устранения. Освещена тематика автоматизации в рассматриваемой сфере информационных технологий. Представлены концептуальные принципы системного и сетевого администрирования. Описаны технологии виртуализации и оверлейные сети. Заключительный раздел отражает проблематику поиска и устранения уязвимостей нулевого дня.

Предназначено для студентов, аспирантов и технических специалистов, которые хотели бы получить базовые знания о системном администрировании и информационной безопасности.

УДК 004.056:004.383.2(075.8)

ISBN 978-5-7782-3484-0

ОГЛАВЛЕНИЕ

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

ИБ – информационная безопасность
ЛВС – локальная вычислительная сеть
КВС – корпоративная вычислительная сеть
СУБД – система управления базами данных
IDS – Intrusion Detection System
IPS – Intrusion Prevention System
OSI – Open Systems Interconnection basic reference model
СОВ – система обнаружения вторжений
СПВ – система предотвращения вторжений
МЭ – межсетевой экран
L2+ – устройство канального уровня модели *OSI*, частично включающее функционал сетевого уровня
L2 – устройство канального уровня модели *OSI*
ОС – операционная система
СКЗИ – средство криптографической защиты информации
ЭЦП – электронная цифровая подпись
ПО – программное обеспечение
БД – база данных
ВМ – виртуальная машина

ВВЕДЕНИЕ

Информационно-коммуникационные технологии приобрели глобальный трансграничный характер и стали неотъемлемой частью всех сфер деятельности государства, общества и личности. Частные компании и государственные учреждения организуют локальные вычислительные сети с коммутацией пакетов на базе стека протоколов *TCP/IP*. Уязвимости стека, операционных систем, средств защиты, программного обеспечения, алгоритмов, методов и протоколов могут быть использованы злоумышленниками для нанесения существенного экономического, социального или политического ущерба. Соответственно вопросы обеспечения информационной безопасности приобретают стратегический характер, что отражено в указах Президента Российской Федерации «Об утверждении приоритетных направлений развития науки, технологий и техники в Российской Федерации и перечня критических технологий Российской Федерации» от 7 июля 2011 г. № 899, «Стратегии развития информационного общества в Российской Федерации на 2017–2030 годы» от 09.05.2017 № 20 и «Доктрине информационной безопасности Российской Федерации» от 5 декабря 2016 г. № 646.

Настоящее пособие является результатом совместной плодотворной работы «Научно-исследовательского института информационно-коммуникационных технологий» и «Новосибирского государственного технического университета». Цель издания – формирование профессиональных навыков и компетенций будущих специалистов в области системного администрирования, неотъемлемой частью которого является теория вычислительных систем и сетей с безопасностью информационных ресурсов. Стоит отметить, что концептуальное понимание изложенного материала служит базисом любой профессии в сфере информационных технологий. От читателей требуется выработка умений по поиску, обработке, систематизации и анализу информации. Необходимо научиться принимать взвешенные, конструктивные и аргументированные решения, в минимальные сроки изучать материал на концептуальном уровне и оперативно решать поставленные задачи.

1. ПРОЕКТИРОВАНИЕ КОРПОРАТИВНОЙ ВЫЧИСЛИТЕЛЬНОЙ СЕТИ, ИНТЕГРАЦИЯ КОМПЛЕКСНОГО МЕЖСЕТЕВОГО ЭКРАНА

Ключевые слова: port security, IP-binding, ACL, VLAN, SNMP, Trunk, STP, VirtualBox, LVM, RAID, Windows, Unix/Linux, NAT, DMZ, Kerio control, Traffic inspector, iptables, FirewallD, SELinux, RDP, SSH, SSL, TLS.

Настоящий раздел посвящен основам проектирования корпоративных вычислительных сетей. Рассматриваются различные варианты установки и настройки сетевого оборудования, обеспечения быстродействия и отказоустойчивости узлов, сетевой информационной безопасности и разграничения доступа к внутренним и внешним ресурсам сети, а также реализации удаленного сетевого доступа.

Все практические задания можно выполнять как в терминальных классах, так и на собственных персональных компьютерах. Во избежание утери данных и негативных последствий от потенциальных ошибок рекомендуется использовать программные продукты виртуализации операционных систем. Одним из возможных решений ее является бесплатное программное обеспечение с открытым исходным кодом, распространяющееся по условиям универсальной общедоступной лицензии (англ. *General Public License*) – *VirtualBox*.

1.1. Проектирование КВС

Для аргументированного выбора серверной операционной системы необходимо задать начальные условия – спроектировать вычислительную сеть предприятия, которое будет выступать заказчиком. Количество рабочих хостов – 100 единиц. При размещении сетевого оборудования не следует забывать приводить ключевые сведения по его настройке.

К примеру, управляемые коммутаторы (*L2+*) могут предоставить широкий спектр возможностей по обеспечению ИБ. Рассмотрим наиболее известные механизмы защиты локальных сетей, используемые (применяемые) в управляемых сетевых устройствах.

1.1.1. Функции управляемого сетевого оборудования

Наиболее распространенными функциями коммутаторов для обеспечения информационной безопасности являются:

1) *port security* – привязка *MAC*-адресов к портам устройства;
2) *IP-binding* – привязка *MAC*-адресов к *IP*-адресам;
3) реализация контроля доступа (*Access Control List*);
4) настройка протокола связующего древа *STP* (англ. *Spanning Tree Protocol*);
5) сегментирование сети на логические («виртуальные») подсети (*Virtual Local Area Network*);
6) управление сетью *SNMP* (англ. *Simple Network Management Protocol*).

Port security представляет собой функцию привязки *MAC*-адресов к портам устройства либо указание числа разрешенных *MAC*-адресов. При несовпадении параметров производится блокировка трафика. Использование данной функции сетевых устройств, в частности коммутаторов, позволяет предотвратить несанкционированное подключение к сети, а также исключить возможность проведения ряда атак, направленных на переполнение таблицы продвижения информационных потоков.

IP-binding представляет собой функцию закрепления связки *MAC*- и *IP*-адресов. В случае несоответствия трафик отклоняется коммутатором. В современных сетевых устройствах производители, как правило, объединяют данную функцию с *port security* и называют ее *IP-MAC-Port Binding*.

Не менее полезными функциями обладает протокол связующего древа *STP*, отключающий избыточные линии связи во избежание петель и задействующий их при обрыве. Расширить пропускную способность канала связи поможет технология агрегирования портов в один логический объект (транк). Пропускная способность при этом суммируется.

Более подробного рассмотрения требуют технологии *Access Control List, Virtual Local Area Network* и протокол *SNMP*.

Access Control List (*ACL*) – списки контроля доступа технологии, являются механизмом фильтрации трафика, выполняемой в соответ-

ствии с заданными критериями, представленными в виде набора текстовых выражений. Списки контроля доступа могут служить для различных целей: выполнение дейтаграммной фильтрации; ограничение доступа к маршрутизатору; указание трафика для выполнения шифрования; определение приоритета обработки трафика; указание сетевых адресов для трансляции и т. д.

ACL разделяют на две основные категории: стандартные списки контроля доступа и расширенные. Стандартные списки дают возможность производить фильтрацию трафика по единственному критерию – адресу отправителя, тогда как расширенные обладают значительно большим списком параметров: адреса отправителя и получателя, *TCP/UDP* порты отправителя и получателя, используемый протокол передачи данных и тип трафика для данного протокола (к примеру, для протокола *ICMP* выполнять фильтрацию только *redirect*-сообщений) и т. д. Возможности расширенных списков контроля доступа могут быть дополнены следующими технологиями:

1) динамические списки (англ. *Dynamic ACL*) позволяют разрешать передачу данных через маршрутизатор на определенный период времени. После выполнения подключения к сетевому устройству и аутентификации пользователя динамический список добавляется к уже существующему расширенному списку. Есть возможность указания временного интервала, по истечению которого динамическая запись удаляется из списка в случае ее неиспользования;

2) рефлексивные списки (англ. *Reflexive ACL*) позволяют осуществлять фильтрацию пакетов на основании информации о сеансах, возникающих внутри маршрутизатора: входящий трафик пропускается только в том случае, если к данному ресурсу ранее обращался какой-либо узел;

3) синхронизируемые списки (англ. *Time-based ACL*) дают возможность управления доступом на основе времени. Для этого необходимо создать временной диапазон, задающий конкретное время дня и/или день недели. С помощью данных списков можно, к примеру, разрешить выход во внешнюю сеть с понедельника по пятницу в рабочее время.

Как стандартные, так и расширенные списки контроля доступа могут задаваться двумя способами: нумерованным и именованным. Разумнее использовать именованные списки, поскольку возможно их построчное редактирование, тогда как в нумерованных списках новые строки могут добавляться только в конец списка.

Для корректной настройки *ACL* необходимо помнить следующее:

1) дейтаграммы обрабатываются строго в соответствии с порядком, в котором заданы условия;

2) при попадании пакета под какое-либо условие его обработка прекращается;

3) каждый список в конце содержит неявную команду *deny any*;

4) списки следует размещать согласно следующим правилам: расширенные – ближе к источнику, стандартные – ближе к получателю;

5) на интерфейс, протокол или направление можно размещать не более одного списка;

6) действие списков контроля доступа не распространяется на трафик, генерируемый маршрутизатором.

Virtual Local Area Network (*VLAN*) представляет собой механизм создания логической топологии сети, не зависящий от ее физической топологии. Функционирование *VLAN* основывается на стандарте *IEEE 802.1Q*. Сетевые устройства, объединенные в одну логическую подсеть, взаимодействуют напрямую на канальном уровне независимо от их реального расположения. Напротив, устройства, подключенные к одному коммутатору, но расположенные в разных логических подсетях, остаются на канальном уровне невидимыми друг для друга.

Применение данной технологии обладает рядом преимуществ:

1) возможностью гибкого разделения сетевых устройств на группы: каждому *VLAN* соответствует своя подсеть; устройства, расположенные в отдалении друг от друга, независимо от этого могут располагаться в одной подсети. *VLAN* значительно облегчает задачу добавления новых устройств или изменения связей между ними;

2) снижением нагрузки на сеть за счет сокращения количества широковещательного трафика: один коммутатор может быть разделен на несколько широковещательных доменов;

3) предотвращением широковещательных штормов в сети;

4) снижением потребления полосы пропускания (в сравнении с случаем одного широковещательного домена);

5) упрощением задачи обеспечения безопасности сети: с помощью *VLAN*-политики и правила безопасности можно применять сразу ко всей подсети, а не к отдельным устройствам.

Предназначение протокола *SNMP* состоит в осуществлении сбора информации о положении в сети Интернет сетевыми управляющими станциями. Формат данных задается протоколом, а выполнение их обработки и интерпретации определяется управляющими станциями.

В основе протокола заложена концепция, что вся необходимая информация для управления устройством хранится на самом устройстве – в управляющей базе данных *MIB* (англ. *Management Information Base*). *MIB* содержит набор переменных, обслуживаемых *SNMP*-агентом, характеризующих состояние управляемого объекта и описывающих его различные параметры: состояние интерфейсов устройства, число обработанных им пакетов, время функционирования и т. д. Кроме того, помимо стандартных переменных каждый производитель может добавлять в управляющую базу данных какие-либо специфичные для данного устройства параметры. Обновление *MIB* регулярно выполняется самим устройством.

SNMP как сетевой протокол представляет собой не что иное, как набор команд для работы с переменными *MIB*. Для осуществления контроля работы сетевого устройства необходимо получить доступ к его *MIB* и проанализировать значения соответствующих переменных. Простота протокола управления сетью достигается за счет отсутствия в нем конкретных команд для управления сетевыми устройствами – вместо этого выполняется переключение переменных *MIB*, что воспринимается узлом как указание к выполнению какой-либо команды. При этом *SNMP* является мощным инструментом, позволяющим стандартизированно задавать наборы команд для управления сетевыми устройствами. Таким образом, обеспечение выполнения команд достигается посредством регистрации переменных управляющей базы данных *MIB* и реакции устройства на их изменение.

1.1.2. Инструменты проектирования

В настоящее время на рынке существует множество инструментов моделирования и проектирования корпоративных вычислительных сетей. Использование подобных средств позволяет осуществить грамотный выбор сетевой топологии, архитектуры и концепции информационной безопасности. Перечислим наиболее распространенные из них:

1) *Cisco Packet Tracer* – программа моделирования сетей, предоставляющая широкий спектр возможностей. Она позволяет экспериментировать с поведением сети, проверять топологии на работоспособность и моделировать различные сценарии развития событий. Пакет включает в себя ряд серий коммутаторов и маршрутизаторов *Cisco*, межсетевой экран *ASA* 5505, серверы *FTP*, *DNS*, *Syslog*, *DHCP* и т. д. Данный инструмент помогает приобретать практические навыки конфигурирования различных протоколов и сетевых устройств коман-

дами *Cisco IOS*. *Cisco Packet Tracer* доступен бесплатно для студентов Сетевой академии *Cisco*;

2) *Microsoft Visio* – инструмент создания диаграмм и блок-схем для ОС *Windows*. С его помощью можно создавать графические отображения различных задач и использовать встроенные шаблоны: задачи управления, разработки приложений, планирования системы безопасности и т. д;

3) онлайн-инструменты проектирования – *draw.io*, *gliffy.com*, *creately.com* – являются удобными решениями, не требующими установки. Они включают наборы элементов для множества типов диаграмм: *UML* (англ. *Unified Modeling Language* – унифицированный язык моделирования), соединения, блок-схемы, инструменты создания структур *iOS* и *Android* приложений и т. д.

Практически каждый производитель сетевого оборудования предлагает программное обеспечение по моделированию и проектированию. Каждое из перечисленных средств обладает как преимуществами, так и недостатками. Выбор конкретного инструмента моделирования зависит от требований к проекту сети и личных предпочтений пользователя.

1.1.3. Практические рекомендации

Одним из наиболее важных компонентов сети является кабельная система. Ошибки, допущенные при ее проектировании, в дальнейшем могут вызывать затруднения в устранении неисправностей. При прокладывании новой проводки кабеля или выполнении обновления уже существующей следует придерживаться стандартов Ассоциации производителей средств связи (англ. *TIA – Telecommunications Industries Association*) и Ассоциации производителей электронного оборудования (англ. *EIA – Electronic Industries Association*). В данных стандартах предусматривается использование экранированной и неэкранированной витой пары, а также оптического кабеля.

Наиболее часто используется *UTP*-кабель (англ. *Unshielded Twisted Pair* – неэкранированная витая пара), обеспечивающий наибольшую гибкость при наименьшей стоимости. Для применения в локальных сетях рекомендуются кабели категории 5, поскольку они обеспечивают поддержку высокоскоростных протоколов. Большая часть высокоскоростных технологий (*Fast Ethernet*, *Gigabit Ethernet*, *FDDI* т. д.) ориентирована на использование кабеля соответствующей категории.

Установка волоконно-оптической системы обходится дороже за счет более высокой стоимости сетевого оборудования, рассчитанного на данный вид соединения. Однако оптоволоконный кабель обладает существенными преимуществами перед витой парой: значительно более высокая скорость передачи данных и возможность передачи их на дальние расстояния. Другим немаловажным аспектом является неподверженность воздействию помех электромагнитной природы. Именно поэтому при необходимости наружной прокладки сети рекомендуется выбирать волоконно-оптический кабель.

Независимо от выбранного типа среды при проектировании кабельной системы необходимо предусмотреть возможность ее расширения – заранее увеличить диаметр каналов для прокладки кабеля, что в дальнейшем позволит легко протянуть дополнительный кабель как *UTP*, так и оптический.

При выборе сетевого оборудования, а именно коммутаторов, необходимо учитывать реализованный в нем способ передачи и буферизации пакетов, а также наличие поддержки технологии *PoE* (англ. *Power over Ethernet*). Технология описывается стандартами *IEEE 802.3af-2003* и *IEEE 802.3at-2009* и позволяет коммутатору *Ethernet* питать конечное устройство (*IP*-камеру, *Wi-Fi* модуль и т.д.) через стандартную витую пару. Коммутаторы, поддерживающие *PoE*, обладают более высокой стоимостью; тем не менее их внедрение позволяет избежать прокладывания дополнительного кабеля к каждому устройству.

1.2. Установка и настройка серверной ОС

При любом серверном решении всегда нужно прогнозировать перспективы дальнейшего развития компании, а соответственно и дальнейшие требования к хосту. В первую очередь речь идет о быстродействии и отказоустойчивости. Решить эту задачу поможет грамотный выбор оборудования и настройка массива из нескольких запоминающих устройств, управляемых контроллером, связанных между собой скоростными каналами передачи данных и воспринимаемых внешней системой как единое целое. Такая технология именуется *RAID*-массивом (англ. *Redundant Array of Independent Disks*). Контроллер может быть представлен как аппаратной, так и программной составляющей. В зависимости от типа используемого массива обеспечивается различная степень отказоустойчивости и быстродействия (скорости чтения/записи данных). Например, *RAID0* – дисковый массив повышен-

ной производительности с чередованием и без отказоустойчивости, а *RAID1* – зеркальный дисковый массив с отказоустойчивостью.

Представим ситуацию: был сконфигурирован файловый сервер на *Linux*, но объем дискового пространства быстро исчерпался. Необходимо вставить дополнительный носитель информации и вновь сконфигурировать сервер для корректной совместной работы с данными по сети. В целях экономии времени можно заблаговременно на этапе установки системы подключить технологию управления дисковым пространством, функционирующую поверх логических разделов (англ. *logical volume manager, LVM*), что даст возможность более гибко использовать дисковое пространство. Основная область применения *LVM* – файловые хранилища, базы данных. Посредством данного инструмента вы объединяете физические носители информации в одну виртуальную логическую единицу, т. е. создаете один виртуальный диск суммарного объема. Затем произвольно разбиваете его пространство. Раздел */boot* не может располагаться в группе логических томов, так как в этом случае загрузчику не удастся его прочитать. Когда же свободное место закончится, нужно будет вставить дополнительный носитель информации и примонтировать его к *LVM*. Никаких дополнительных настроек не потребуется. Аналогичными функциями (но ограниченными) обладает инструмент Windows под именем «Управление дисковыми пространствами».

После успешной установки ОС необходимо заняться вопросами информационной безопасности, в первую очередь – межсетевым экраном (фаерволом/брандмауэром). Среди решений с открытым исходным кодом оптимальным является пакетный фильтр *IPtables*, входящий в состав встроенного в ядро *Linux* межсетевого экрана *Netfilter* и осуществляющий фильтрацию трафика и перенаправление пакетов в соответствии с заданными правилами. Стоит отметить, что в свежих версиях *Linux CentOS* и *RedHat* на замену *Netfilter* (в том числе *IPtables*) пришел *FirewallD*. Однако отказ от применения *FirewallD* не оказывает негативного влияния на безопасность ЛВС, поскольку функционал и возможности *IPtables* позволяют не менее качественно реализовать фильтрацию входящего, исходящего и транзитного трафика. Приступая к его настройке, рекомендуется тщательно ознакомиться с руководством под названием *Iptables Tutorial*. Пример настройки пакетного фильтра с трассировщиком соединений *Linux*/*Unix* поможет концептуальному пониманию материала. Далее необходимо провести сравнительный анализ возможных реализаций интернет-шлюза и совместимых средств защиты [1–16].

1.3. Настройка межсетевого экрана

Остановимся на технологии выполнения преобразований сетевых адресов *NAT* (англ. *Network Address Translation*). Отметим, что только первый пакет из потока проходит через цепочки этой таблицы, трансляция адресов или маскировка применяется ко всем последующим пакетам в потоке автоматически. Действие *DNAT* (англ. *Destination Network Address Translation*) производит преобразование адресов назначения в заголовках пакетов. Другими словами, этим действием производится перенаправление пакетов на другие адреса, отличающиеся от указанных в заголовках пакетов. *SNAT* (англ. *Source Network Address Translation*) используется для изменения исходных адресов пакетов. С помощью этой процедуры можно скрыть структуру локальной сети, а вместе с этим разделить единственный внешний *IP*-адрес между компьютерами локальной сети для выхода в Интернет. В этом случае брандмауэр с помощью *SNAT* автоматически производит прямое и обратное преобразование адресов, тем самым давая возможность выполнять подключение к серверам в Интернете с компьютеров в локальной сети.

Рассмотрим пример настройки пакетного фильтра *iptables*. Существует два подхода к ее осуществлению: запрет по умолчанию на все, что не попадает под действие ни одного из правил, либо разрешение всех действий, не запрещенных ни одним правилом. В данном примере использован первый принцип:

```
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT
```

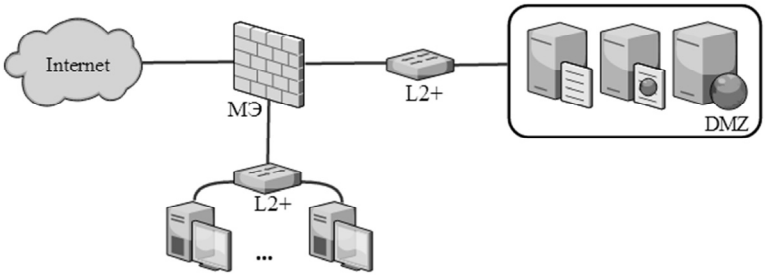
Затем создается новая цепочка правил *«bad_tcp_packets»* для обработки *TCP*-пакетов с некорректными заголовками:

```
iptables -N bad_tcp_packets
iptables -A bad_tcp_data -p tcp --tcp-flags SYN,ACK SYN,ACK -m state --state NEW -j REJECT --reject-with tcp-reset
iptables -A bad_tcp_packets -p tcp! --syn -m state --state NEW -j LOG --log-prefix "New not syn:"
iptables -A bad_tcp_packets -p tcp! --syn -m state --state NEW -j DROP
iptables -A INPUT -p tcp -j bad_tcp_packets
```

Далее создается цепочка «*allowed*» для обработки *TCP*-пакетов:

```
iptables -N allowed
iptables -A allowed -p TCP --syn -j ACCEPT
iptables -A allowed -p TCP -m state --state
ESTABLISHED,RELATED -j ACCEPT
```

Для разграничения доступа к внешним и внутренним ресурсам сети можно использовать демилитаризованную зону (англ. *DMZ – Demilitarized Zone*). Ее настройка осуществляется либо с помощью встроенных функций сетевого оборудования, либо с использованием технологии *DNAT* (см. рисунок) совместно с сегментацией трафика *VLAN*.



Настройка демилитаризованной зоны

Создание демилитаризованной зоны позволяет минимизировать ущерб в случае атаки на какой-либо из общедоступных сервисов: у злоумышленника будет отсутствовать прямой доступ во внутренний сегмент сети.

1.4. Реализация удаленного сетевого доступа

При организации удаленного доступа к информационной системе не следует забывать производить фильтрацию по *IP*-адресам. К тому же недопустимо в стандартном виде использовать *TELNET* (англ. *TErminaL NETwork*), *SSH* (англ. *Secure Shell*) и *RDP* (англ. *Remote Desktop Protocol*) протоколы. К примеру, в *RDP* стоит задействовать *TLS* (англ. *Transport Layer Security*), ограничить количество неправильных попыток ввода паролей, инициализировать проверку параметров и прохождение аутентификации на сетевом уровне, повысить уровень шифрования *ACL* и т. д. С практической точки зрения использование данных протоколов внутри защищенного канала является наиболее содержа-

тельным и безопасным решением: например, ввести в действие инкапсуляцию информационных потоков в *VPN* (англ. *Virtual Private Network* – виртуальная частная сеть). Наиболее известными реализациями данных технологий являются протоколы *IPsec* (англ. *IP Security*), используемые, как правило, в паре с протоколом *L2P* (англ. *Layer 2 Tunnel Protocol*) для обеспечения безопасности передаваемых данных, и *OpenVPN*. Более подробно данные протоколы рассматриваются в разделе 9 настоящего пособия. Не стоит также забывать и о технологии простукивания портов (англ. *Port Knocking*), представляющей собой схему аутентификации и авторизации через заданную последовательность попыток соединения с портами сервера. Целевой порт открывается после серии заранее определенных «стуков», фиксируемых на стороне сервера. Данная схема является уязвимой в случае прослушивания трафика злоумышленником, поэтому в ней предусмотрен механизм использования одноразовых последовательностей.

После реализации проекта корпоративной вычислительной сети и виртуального внедрения сконфигурированного сервера в головной офис стоит проанализировать возможные последствия и процессы при проведении следующих видов атак:

1) ложные *ARP*-ответы. Производится рассылка сфальсифицированных *ARP*-сообщений таким образом, что каждый из атакуемых хостов интерпретирует *MAC*-адрес злоумышленника адресом своего собеседника;

2) навязывание ложного маршрутизатора. Навязывание ложных маршрутов выполняется с помощью фальсифицированных *ICMP*-сообщений *Redirect*. Адрес злоумышленника становится известным в качестве адреса маршрутизатора;

3) подлог при конфигурировании хоста. Возможно навязывание ложного пограничного узла посредством *ICMP*-сообщения *Router Advertisement*; альтернативный вариант – через протокол динамической настройки узла *DHCP* (англ. *Dynamic Host Configuration Protocol*) с помощью выдачи первого *DHCP*-предложения подставным сервером;

4) воздействие на протоколы маршрутизации. Для переключения требуемых маршрутов на собственный сетевой узел рассылаются фальсифицированные сообщения протоколов маршрутизации с более выгодными параметрами (например, меньшим числом хопов – транзитных узлов);

5) имперсонация (выдача себя за другой объект) без обратной связи – достигается изменением содержимого заголовков дейтаграмм;

6) десинхронизация *TCP*-соединения. Осуществляется рассылкой фальсифицируемых управляющих сообщений. К этой же категории можно отнести атаки туннелирования и обхода правил фильтрации например, с помощью двойной инкапсуляции пакетов;

7) имперсонация с целью установления полного контроля над соединением в англоязычной литературе именуется *TCP hijacking;*

8) использование уязвимостей протоколов прикладного уровня.

Аналитические выкладки по данным вопросам помогут более тщательно подойти к обеспечению сетевой информационной безопасности. Необходимо отметить, с какими видами атак реализованный комплекс мер обеспечения ИБ все-таки не смог справиться, и предложить алгоритм нейтрализации данных угроз.

1.5. Практическое задание

Необходимо спроектировать корпоративную вычислительную сеть, состоящую из 100 хостов. При построении распределенной филиальной инфраструктуры ключевые серверы необходимо спрятать за *NAT* главного офиса. При размещении сетевого оборудования (коммутаторы 2+, маршрутизаторы, межсетевые экраны и др.) следует задокументировать ключевые настройки с пояснением целесообразности использования различных протоколов, технологий и интеллектуальных функций.

Далее требуется установить и сконфигурировать головной шлюз на базе серверной операционной системы (ОС) *Windows* или *Unix*/*Linux*. В сравнительном анализе аргументировать выбор не только семейства операционных систем, но и конкретного продукта. Ключевые параметры сравнения: надежность/отказоустойчивость, информационная безопасность, быстродействие, трудоемкость сопровождения.

При первоначальном конфигурировании сервера стоит обратить внимание на следующие аспекты:

1) резервное копирование и масштабируемость;

2) организация и защита удаленного доступа (протоколы, технологии и алгоритмы аутентификации и др.);

3) настройка межсетевого экрана (включая *NAT* и *DMZ*);

4) устранение опубликованных уязвимостей и известных инструментов скрытого управления компьютером (бэкдоров);

5) установка дополнительных средств защиты.

В ходе выполнения практического задания возникнет ряд интересных вопросов. Стоит ли устанавливать антивирус на интернет-шлюз? Если да, то какой? Удовлетворяют ли интегрированные в ОС решения по реализации *NAT* с точки зрения информационной безопасности?

Содержание отчета

1. Цель работы.
2. Техническое задание.
3. Сравнительный анализ существующих решений.
4. Установка и конфигурирование серверного решения.
5. Реализация комплекса мер по обеспечению ИБ.
6. Заключение.

Контрольные вопросы

1. Какие возможности обеспечения ИБ реализованы в управляемых сетевых устройствах? Приведите примеры.
2. Каков принцип работы списков контроля доступа? В чем состоят различия стандартных и расширенных списков? Приведите примеры.
3. Каким образом применение технологии *VLAN* позволяет уменьшить количество широковещательного трафика в сети?
4. Изложите принцип работы простого протокола управления сетью *SNMP*.
5. Приведите пример корректной настройки дискового пространства. Как можно обеспечить отказоустойчивость и быстродействие сервера?
6. Каков принцип работы пакетного фильтра *IPtables*? Что представляет собой трассировщик соединений?
7. Как применяются технологии *DNAT* и *SNAT*?
8. Для каких целей служит создание в сети демилитаризованной зоны (*DMZ*)? Какие опасности это может повлечь за собой?
9. Как предпочтительнее организовать удаленный сетевой доступ? Ответ обоснуйте.
10. Перечислите наиболее распространенные сетевые атаки, их возможные последствия и методы противодействия.

2. ВЕБ-СЕРВЕРЫ И МЕЖСЕТЕВЫЕ ЭКРАНЫ УРОВНЯ ВЕБ-ПРИЛОЖЕНИЙ

Ключевые слова: веб-сервер, СУБД, серверный обработчик событий, DNS, HTTP, HTTPS, SSL, TLS, DDOS, SQL-Injection, XSS, Flash, JavaScript, PDF, XML, IFRAME, CSRF, brute force, фишинг.

Раздел посвящен рассмотрению наиболее распространенных угроз безопасности веб-ресурсов и способам их нейтрализации. Детально раскрываются распределенные атаки на отказ в обслуживании и различные их вариации.

2.1. Веб-серверы

Веб-сервером называют программное обеспечение, основной функцией которого является получение и обработка запросов от клиентов. С точки зрения аппаратного обеспечения веб-сервер представляет собой компьютер, хранящий файлы сайта и доставляющий их на устройство конечного пользователя. В качестве клиента обычно выступает веб-браузер: он передает веб-серверу запросы на получение каких-либо ресурсов (*HTML*-страниц, файлов, изображений и т. д.), указывая их *URL* (англ. *Uniform Resource Locator*), и в ответ получает запрашиваемые данные. Обмен данными осуществляется с помощью протокола *HTTP* (англ. *HyperText Transfer Protocol*) либо *HTTPS* (англ. *HyperText Transfer Protocol Secure*). Различают статический и динамический веб-контенты. Статический содержит неизменную текстовую и мультимедийную информацию, тогда как динамический генерируется в зависимости от действий пользователя.

Среди функций веб-серверов выделяют:

1) автоматизацию работы веб-страниц;

2) поиск обновлений баз данных для различных программ (антивирусов, торрент-клиентов и т. д.);

3) аутентификацию и авторизацию пользователей;

4) ведение журнала обращений к каким-либо ресурсам.

Рассмотрим наиболее распространенные угрозы безопасности веб-серверов.

2.2. Основные угрозы безопасности веб-ресурсов

При техническом сопровождении веб-сервера на практике приходится сталкиваться со следующими видами злоумышленных воздействий [17–36]:

1) распределенные атаки на отказ в обслуживании (англ. *Distributed Denial of Service, DDoS*) – доведение вычислительной системы до отказа посредством многочисленных распределенных запросов;

2) *SQL*-инъекции. Представляют собой атаки, направленные на веб-приложения и позволяющие модифицировать логику выполнения *SQL*-запросов путем внедрения в них произвольного *SQL*-кода;

3) межсайтовый скриптинг (англ. *XSS – Cross-Site Scripting*) – атака, использующая уязвимости скриптовых языков на стороне клиента (*JavaScript*, *Flash*), производится путем подстановки вредоносного кода в генерируемую сервером страницу;

4) атаки, основанные на вводе некорректных данных форм. В случае настройки недостаточно строгих или некорректных проверок вводимых данных злоумышленник получает возможность отправки собственных данных. Это позволит ему, к примеру, заполнить базу данных недействительными записями и далее произвести *DDoS*-атаку, запрашивая страницу, отображающую несуществующие данные;

5) эксплуатация недоработок в коде информационного ресурса;

6) недостаточное противодействие автоматизации – возможность автоматического выполнения операций, которые следует выполнять вручную. Стандартным примером являются системы автоматизированного поиска уязвимостей;

7) модификация файлов *cookie*;

8) похищение сессий;

9) модификация *URL*;

10) атаки, основанные на «медленных» *HTTP*-запросах (*slow HTTP Post* и *slow HTTP Headers*) – медленная передача на сервер маленькими частями *POST*-запросов и *HTTP*-заголовков. Соединение не закрывается до окончания передачи данных. Большое количество подобных соединений приводит к истощению ресурсов сервера и его перегрузке;

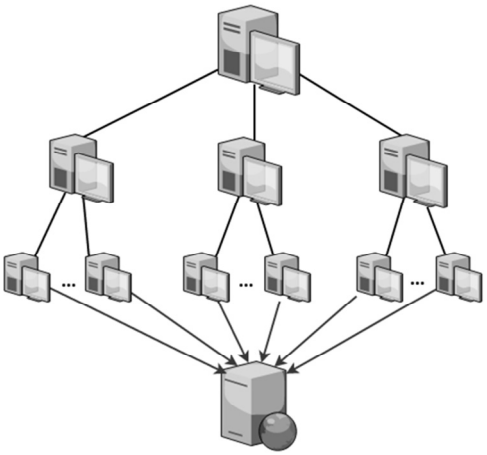
11) эксплуатация уязвимостей в криптографических алгоритмах;

12) эксплуатация уязвимостей веб-сервера и ОС.

Более подробного рассмотрения требует один из самых известных видов атак – распределенные атаки на отказ в обслуживании. Этот вид атак является наиболее распространенным и эффективным, поскольку

гарантированных методов защиты от них не существует, а специальных знаний для проведения *DDoS*-атак не требуется.

Структура *DDoS* является иерархической и состоит, как правило, из трех уровней: управляющая консоль, промежуточные машины и «зараженные» агенты (см. рисунок).



Структура DDoS-атаки

Управляющая консоль подает сигнал о начале атаки, далее промежуточные компьютеры передают сигнал агентам, а те в свою очередь посылают запросы на атакуемый узел. Использование подобной структуры атаки практически полностью исключает возможность обнаружения ее инициатора.

Перечислим основные виды *DDoS*-атак.

SMURF – рассылка сфальсифицированных *Echo*-запросов от имени жертвы по широковещательному адресу сети. Все узлы, получившие такой запрос, отправляют *Echo*-ответ узлу, инициировавшему атаку, в результате чего вся сеть подвергнется перегрузке из-за шквала ответных сообщений и соответственно отказу в обслуживании.

SYN Flood – рассылка многочисленных запросов на *TCP*-соединение с атакуемых хостов. Приводит к истощению ресурсов, выделяемых им для обслуживания будущих соединений. В результате атакованный узел в заблокированном состоянии в течение длительного времени не способен принимать новые запросы на установление соединения.

UDP Flood – отправка множества *UDP*-сообщений на атакуемый хост. Приводит к увеличению нагрузки на канал связи и перегрузке атакуемой системы.

Ping-of-Death – посылка атакуемому хосту фрагментированного *ICMP*-пакета размером, превышающим 64 Кбайта, что может привести к переполнению сетевого стека и переходу узла в недееспособное состояние.

Land – посылка на атакуемый узел *SYN*-пакета с идентичными адресами отправителя и получателя, в результате которой узел попадает в петлю бесконечных обращений к самому себе.

WinNuke – посылка на атакуемый узел срочных служебных данных (англ. *Out of Band*, *OOB*) для *TCP*-соединения через порт 139. Приводит к крушению системы (так как в *Windows*-системах не предусмотрена возможность приема срочных данных).

Teardrop – атака, основанная на использовании ошибки, возникающей при сборке датаграммы (ошибка накладывающихся *IP*-фрагментов). На атакуемый узел отправляется датаграмма с ложными значениями начала и длины фрагмента, что после ее сборки вызывает ошибки памяти и крушение *Windows*-системы.

Стандартными механизмами противодействия *DDoS*-атакам являются применение капчи (англ. *CAPTCHA – Completely Automated Public Turing test to tell Computers and Humans Apart*), введение лимита на число одновременных подключений, лимита на время подключения инициализации черных списков.

Необходимо самостоятельно раскрыть детали вышеприведенных атак и исследовать методы противодействия им, продемонстрировав навыки поиска, обработки и анализа информации в глобальной сети Интернет.

2.3. Практическое задание

Требуется произвести сравнительный анализ востребованного на рынке программного обеспечения для реализации веб-сервера. На базе развернутого серверного решения в практических заданиях к первому разделу (или отдельной виртуальной машины) необходимо установить выбранный комплекс продуктов. Далее произвести его конфигурирование с установкой дополнительных средств защиты, нейтрализующих ранее описанные угрозы. Не забывайте, что в ключевых словах к каждой главе всегда приводятся подсказки.

Содержание отчета

1. Цель работы.
2. Техническое задание.
3. Сравнительный анализ существующих решений.
4. Установка и конфигурирование серверного решения.
5. Реализация комплекса мер по обеспечению ИБ.
6. Заключение.

Контрольные вопросы

1. Что представляет собой веб-сервер?
2. С какими угрозами безопасности может столкнуться администратор веб-сервера?
3. Чем опасны «медленные» *HTTP*-запросы?
4. Назовите стандартные способы противодействия *SQL*-инъекциям.
5. Какова классическая структура *DDoS*-атаки?
6. Перечислите существующие виды *DDoS*-атак и методы противодействия им.
7. Как реализуется межсайтовый скриптинг?
8. Приведите пример межсайтовой фальсификации запроса.
9. Почему не следует открывать *PDF*-файлы в браузере?
10. Какие инструментальные средства защиты позволят нейтрализовать описанные угрозы?

3. ФАЙЛОВЫЕ СЕРВЕРЫ

Ключевые слова: файловый сервер, многопользовательский доступ, FTP, FTPS, SFTP, p2p, cloud, CrushFTP Server, Internet Information Services, WebDAV, NAS, SAN, BitTorrent.

Настоящий раздел посвящен изучению технологий централизованного и децентрализованного файлообмена. Рассматриваются различные способы организации хранения данных, анализируются их преимущества и недостатки.

3.1. Технологии централизованного файлообмена

Под файловым сервером понимается ресурс, предназначенный для выполнения файловых операций ввода-вывода и хранящий файлы любого типа.

Этот сервер требователен к объему дискового пространства и скорости доступа. Соответственно необходимо использовать технологии *RAID*-массивов и *LVM* для обеспечения бесперебойной работы и повышенной скорости записи и чтения данных.

При выборе дисков не стоит забывать следующие аспекты.

1. *SATA* (англ. *Serial ATA*) – последовательный интерфейс обмена данными с накопителями информации. Целевая группа – персональные компьютеры и бюджетное серверное оборудование. По сравнению с *SAS*- и *SSD*-дисками скорость чтения и записи *SATA*-дисков заметно ниже. Однако достоинством является большой объем хранимой информации.

Диски *SATA* хорошо подойдут для серверов, работа которых не требует частой записи и чтения информации. Например, потоковые операции:

- ✓ кодирование видео;
- ✓ хранилища данных;
- ✓ системы резервного копирования;
- ✓ файловые серверы большого объема с малой параллельной нагрузкой.

2. *Serial Attached SCSI* (*SAS*) – последовательный компьютерный интерфейс, разработанный для подключения различных устройств хранения данных (жестких дисков и ленточных накопителей). *SAS* разработан для замены параллельного интерфейса *SCSI* и использует тот же набор команд.

Обеспечение высокой надежности хранения данных *SAS* обусловливает их целевое назначение:

- ✓ хостинг;
- ✓ системы управления базами данных (СУБД);
- ✓ веб-серверы с высокой нагрузкой;
- ✓ распределенные системы;
- ✓ системы, обрабатывающие большое количество запросов (терминальные и 1С-серверы).

Главным недостатком *SAS*-дисков (аналогично *SSD*) является их небольшой объем и высокая цена.

3. Твердотельный накопитель (англ. *Solid-State Drive, SSD*) – компьютерное немеханическое запоминающее устройство на основе микросхем памяти с управляющим контроллером. По сравнению с традиционными жесткими дисками (*HDD* и *SATA*) твердотельные накопители имеют меньший размер и вес, но практически в пять раз

большую стоимость за гигабайт и значительно меньшую износостойкость (ресурс записи). Стоит перечислить типы памяти по возрастанию скоростных и стоимостных показателей: *MLC* (англ. *Multi-Level Cell*), *TLC* (англ. *Triple-Level Cell*) и 3D *NAND*.

В *SSD*-дисках нет динамических звеньев, что обеспечивает высокую механическую стойкость, сниженное энергопотребление и высокую скорость работы. В данный момент *SSD*-диски обеспечивают максимально возможную скорость чтения и записи, что позволяет использовать их практически для любых высоконагруженных проектов.

Наиболее популярные форм-факторы: 2.5", *mSATA*, M.2, *PCI-E*. Стоит также выделить новый логический интерфейс *NVM* Express, разработанный специально для твердотельных накопителей. От старого *AHCI* он отличается более низкими задержками доступа и высокой параллельностью работы чипов памяти за счет нового набора аппаратных алгоритмов.

Под каждый серверный объект можно выбрать защищенную вариацию *FTP*-сервера, который бы удовлетворял вашему проекту корпоративной вычислительной сети. На рынке представлен широкий спектр программного обеспечения: *wu-ftpd, ProFTPD, Pure-FTPd, SlimFTPd, vsftpd, Internet Information Services, glFTPd, CrushFTP Server, GoAnywhere Services, Cerberus FTP Server, FileZilla Server* и др.

Допустимым решением может выступить технология *NAS* (англ. *Network Attached Storage*) – сетевая система хранения данных, сетевое хранилище. Представляет собой производительный сервер или кластер серверов с дисковым массивом, подключенным к сети и поддерживающим работу по принятым в ней протоколам. Естественно, присутствует поддержка *RAID*-массивов. Для *NAS* характерна надежность хранения данных, легкость доступа для пользователей, легкость администрирования, масштабируемость.

Стоит различить подобную реализацию с технологией *SAN* (Сеть хранения данных, СХД – англ. *Storage Area Network*), которая представляет собой архитектурное решение для подключения внешних устройств хранения данных таким образом, чтобы операционная система распознала подключенные ресурсы как локальные.

3.2. Технологии децентрализованного файлообмена

Альтернативным вариантом может послужить *BitTorrent* (букв. англ. «битовый поток») – пиринговый (децентрализованный, *P2P*) сетевой протокол для кооперативного обмена файлами через глобальную

сеть Интернет. Файлы передаются блоками, каждый *torrent*-клиент, получая (скачивая) эти части, в то же время отдает (закачивает) их другим клиентам, что снижает нагрузку и зависимость от каждого клиента-источника и обеспечивает избыточность данных.

Рассмотрим более подробно принцип работы данного протокола.

Для выполнения скачивания файла клиент соединяется с торрент-трекером (англ. *torrent tracker*) – сервером, связывающим клиентов *BitTorrent* друг с другом, и передает свой *IP*-адрес и хэш-сумму файла для скачивания. Далее он получает от трекера *IP*-адреса прочих клиентов, качающих или раздающих торрент-файл; в процессе скачивания список *IP*-адресов регулярно обновляется. Данные между клиентами передаются без прямого участия торрент-трекера: он занимается лишь сбором информации о процессе скачивания, подключенных клиентах и т. д. После установления соединения клиенты обмениваются информацией о сегментах файла, хранящегося у них. Клиент, желающий скачать файл, посылает запрос на скачивание, получает сегмент в случае готовности второго клиента и затем сравнивает контрольную сумму фрагмента с суммой, записанной в торрент-файле. В случае совпадения сумм скачивание считается успешно выполненным и клиент оповещает остальных участников о наличии у него фрагмента файла.

В целях оптимизации раздачи клиент может приостановить передачу фрагментов торрент-файла другому клиенту: приоритет присваивается тому узлу, который сам передал наибольшее число сегментов.

Использование данной технологии обладает рядом преимуществ:

1) отсутствуют очереди на скачивание;

2) скачанные фрагменты мгновенно становятся доступны другим пользователям;

3) производится контроль целостности каждого фрагмента;

4) объектом раздачи могут служить несколько файлов (к примеру, содержимое каталога);

5) высокая скорость скачивания, растущая с увеличением числа клиентов.

В более новых версиях протокола предусмотрена возможность выполнения раздачи файлов без трекера с помощью распределенной хэш-таблицы *DHT* (англ. *Distributed Hash Table*), что решает проблему отказа работы всей сети при отказе трекера. Работа без торрент-трекера возможна также при использовании мультипротокольных клиентов с поддержкой *BitTorrent*, к примеру, клиент *Shareaza* выполняет обмен *IP*-адресами узлов других поддерживаемых сетей, в том числе *BitTorrent*, через сеть *Gnutella2*.

Помимо *BitTorrent* существует множество других реализаций файлообменных сетей: *Direct Connect*, *Gnutella*, *Gnutella2*, *FastTrack* и т. д.

Не стоит исключать из анализа и облачное хранилище данных (англ. *cloud storage*) – модель онлайн-хранилища, в котором данные хранятся на многочисленных распределенных в сети серверах, сконфигурированных с использованием технологий виртуализации.

После выбора технологии файлообменного сетевого доступа необходимо будет проработать корректный многопользовательский режим работы. Следующий важный этап – тестирование объектов в срезе информационной безопасности.

3.3. Практическое задание

Прежде всего необходимо овладеть теоретической частью материала [37–41] и изучить существующие технологии файлообменного сетевого доступа. Затем выполнить поиск и сравнительный анализ существующих решений (как бесплатных, так и проприетарных). Развернуть наиболее рентабельное решение на прежней или новой виртуальной машине или представить проектом. Установить дополнительные модули защиты от внешних злоумышленных воздействий. Произвести аудит ИБ развернутого серверного решения. Проиллюстрировать корректный многопользовательский режим работы. Проанализировать возможные уязвимости объекта.

Содержание отчета

1. Цель работы.
2. Техническое задание.
3. Сравнительный анализ существующих технологий.
4. Сравнительный анализ существующих решений.
5. Реализация выбранного проекта.
6. Тестирование серверного решения.
7. Реализация комплекса мер по обеспечению ИБ.
8. Заключение.
9. Список литературы.

Контрольные вопросы

1) Что представляет собой файловый сервер? Какими преимуществами обладает?

2) Для каких серверов предпочтительнее выбирать диски *SATA, SAS, SSD*?

3) Назовите известные вам реализации *FTP*-серверов. Приведите их сравнительный анализ.

4) Что собой представляет технология *NAS*? В чем состоит ее отличие от *SAN*?

5) Изложите принцип функционирования пиринговых (*P2P*) сетей.

6) Какими преимуществами обладает технология *P2P* перед технологиями централизованного файлообмена?

7) Каков принцип работы протокола *BitTorrent*?

8) Перечислите известные вам реализации файлообменных сетей помимо *BitTorrent*. Приведите их существенные различия.

9) В чем состоят особенности облачного хранилища данных?

10) Какие уязвимости имеют рассмотренные технологии организации хранения данных?

4. ОСНОВЫ АДМИНИСТРИРОВАНИЯ ЦЕНТРАЛИЗОВАННОЙ РАБОТЫ ПОЛЬЗОВАТЕЛЕЙ

Ключевые слова: 1С-предприятие, сервер терминалов, виртуализация приложений, толстый и тонкий клиенты, веб-интерфейс, СУБД, MySQL, MS SQL, PostgreSQL, web-сервер, ssl, https, сервер терминалов, Remote Desktop Services, RDMS, Application Virtualization, Active Directory, DNS, DHCP, VMware Horizon, VMware View, gold-iso, PCoIP, сервер печати, screwdrivers.

Раздел посвящен технологиям обеспечения централизованной работы пользователей с приложениями. Излагается материал относительно виртуализации, сервера терминалов, контроллера доменов.

В качестве примера наиболее популярного в России продукта, для которого необходимо обеспечение централизованной работы пользователей, рассмотрим 1С-предприятие – программный продукт компании «1С», предназначенный для автоматизации деятельности на предприятии. Способен работать через толстый и тонкий клиенты, а также веб-интерфейс. В первом варианте обработка информации производится преимущественно на локальной стороне взаимодействия, а при работе с тонким клиентом или веб-интерфейсом – на серверной. Последний режим работы не требует больших ресурсов как системы, так

и канала связи. При этом не стоит забывать о безопасности взаимодействия. Например, вместо протокола *http* следует использовать *https*.

Вариантов построения системы централизованной работы пользователей с приложениями достаточно много [42–45]. Современные стандарты информационных технологий позволяют отказаться от стационарных компьютеров. Ведь установка, сопровождение, обновление программного обеспечения на каждом системном блоке – задача трудоемкая. На сегодняшний день вычислительные мощности значительно опережают программные требования. Это и породило эру виртуализации. Рассмотрим основные варианты решения поставленной задачи: сервер терминалов, виртуализация приложений, *Active Directory*.

4.1. Сервер терминалов

Сервер терминалов (англ. *terminal server*) – сервер, предоставляющий клиентам вычислительные ресурсы (процессорное время, память, дисковое пространство) для решения задач. Технически терминальный сервер представляет собой очень мощный компьютер (вычислительный кластер), соединенный по сети с терминальными клиентами. К последним предъявляются минимальные системные требования. Терминальный сервер служит для удаленного обслуживания пользователя или администратора с предоставлением рабочего стола или консоли.

В терминах корпорации *Microsoft* терминальный сервер именуется *Remote Desktop Services* или *terminal server* в зависимости от версии ОС. Клиенты получают доступ к рабочему столу *Windows Server* или приложению (*RemoteApp*) с использованием протокола удаленного рабочего стола (*RDP*, англ. *Remote Desktop Protocol*), *http*, *https*. Свыше сотни пользователей могут работать на одном терминальном сервере одновременно и изолированно. При таком подходе наблюдается экономия вычислительных ресурсов, выделяемых на одного пользователя, в сравнении с полной виртуализацией отдельных операционных систем (*VDI RDS,* англ. *Virtual Desktop Infrastructure*).

4.2. Виртуализация приложений

Существует технология виртуализацией приложений. Например, *Microsoft Application Virtualization* (*App-V*) позволяет каждому приложению работать в собственной автономной виртуальной среде на

клиентском компьютере. Виртуализированные приложения изолированы друг от друга. Это позволяет избежать конфликтов между приложениями, но они по-прежнему могут взаимодействовать с клиентским компьютером.

Подобные технологии далеко не новшество, более десятка лет назад рынок виртуализации успешно осваивал продукт *Citrix XenApp*. Однако на сегодняшний день лидером является компания *VMware*, *Inc*.

Эта компания ввела в обиход термин *VDI – Virtual Desktop Infrastructure* для позиционирования своей технологии виртуализации персональных компьютеров. Корпорация также занимается виртуализацией приложений. *VMware* реализовало продукты *View*, *ThinApp* – средства для создания инфраструктуры виртуальных ПК предприятия и виртуализации приложений с широким функционалом в области автоматической развертки и сопровождения.

Развитие получил проект *VMware Horizon* – интегрированное решение, которое обеспечивает доступ к опубликованным приложениям и рабочим столам на базе единой платформы. При помощи *Horizon* корпоративные приложения и операционные системы управляются централизованно.

Следует отметить следующие технические аспекты *Horizon*:

1) доступ к опубликованным приложениям и виртуальным рабочим столам на базе единой платформы предлагает упрощенное управление, предоставление прав конечным пользователям и быструю доставку опубликованных приложений, настольных компьютеров, удаленных и виртуальных рабочих столов на различные устройства и места;

2) унифицированное рабочее пространство для упрощенного доступа – конечные пользователи могут получить доступ ко всем приложениям и рабочим столам из единой унифицированной рабочей области;

3) оптимизация систем хранения и доставка из программно-определяемого центра обработки данных;

4) замкнутый цикл управления и автоматизации. Обеспечение контроля рабочего состояния и мониторинга рисков, проактивного мониторинга поведения конечных пользователей и глубокой диагностики от центра обработки данных до устройства внутри единой консоли;

5) централизованное управление образами виртуальных, физических и личных устройств сотрудников;

6) доступ к гибридному облаку.

4.3. Служба каталогов *Active Directory*

Не менее интересным проектом является *Active Directory* («Активный каталог», *AD*), *LDAP* – совместимая реализация службы каталогов корпорации *Microsoft* для операционных систем семейства *Windows NT*. *Active Directory* позволяет администраторам:

1) использовать групповые политики для обеспечения единообразия настройки пользовательской рабочей среды;

2) разворачивать программное обеспечение на множестве компьютеров через групповые политики или посредством *System Center Configuration Manager* (ранее *Microsoft Systems Management Server*);

3) устанавливать обновления операционной системы, прикладного и серверного программного обеспечения на всех компьютерах в сети с использованием службы обновления *Windows Server*.

Active Directory (*AD*) хранит данные и настройки среды в централизованной базе данных. Сети *AD* могут быть различного размера: от нескольких десятков до нескольких миллионов объектов. Сетевые ресурсы, о которых хранится информация в базе данных, в терминах *Active Directory* называются объектами и являются отдельными именованными наборами атрибутов. Атрибуты зависят от типа объекта и представляют собой характеристики и данные, которые могут в нем содержаться (рис. 4.1).

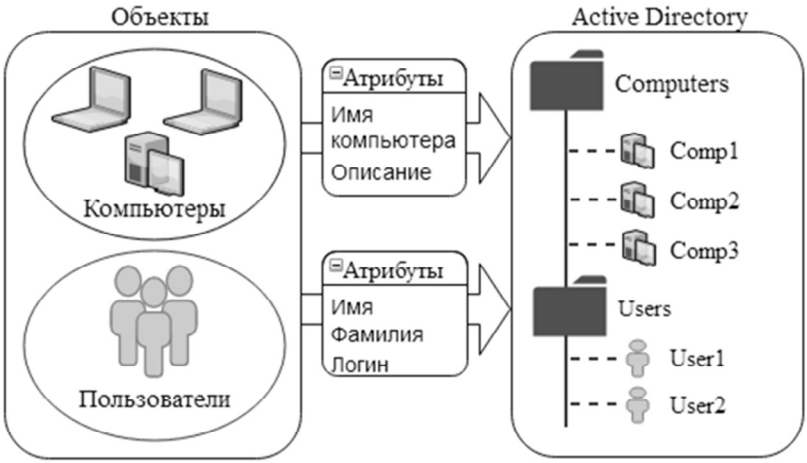


Рис. 4.1. Схема объектов Active Directory

Active Directory можно рассматривать с точки зрения логической и физической структуры. Логическая структура определяет множество объектов, которые могут храниться в каталоге: для каждого класса объектов задаются обязательные и дополнительные атрибуты его представителей, а также то, какой класс объектов может являться родительским по отношению к данному.

Логическая структура включает следующие компоненты:

1) организационные подразделения – логические контейнеры, позволяющие группировать объекты;

2) домены – базовая структурная единица *Active Directory*;

3) деревья доменов – система доменов, имеющая иерархическую структуру и единый корень (корневой домен);

4) леса доменов – множество деревьев доменов, находящихся в какой-либо форме доверительных отношений.

Иерархия логических компонентов *AD* представлена на рис. 4.2.

Рис. 4.2. Логическая структура Active Directory

Поскольку логическая структура *AD* не зависит от физического расположения серверов и сетевых соединений в домене, при ее планировании иерархия доменов определяется независимо от требований физической сети с учетом только административных и организационных требований.

Физическая структура *AD* включает в себя узлы (сайты) и контроллеры доменов и отражает физическую структуру организации. Целью планирования физической структуры *AD* является оптимизация репликации – процесса копирования на все контроллеры изменений, выполненных на каком-либо одном из контроллеров домена. Сайт представляет собой часть сети, в которой все контроллеры домена связаны высокоскоростными линиями связи. Соединения между самими сайтами являются более медленными. Подобная структура объясняется необходимостью частой репликации внутри сайтов с возможностью передачи больших объемов данных без сжатия, тогда как между сайтами этот процесс производится реже, а передаваемые данные необходимо сжимать (рис. 4.3).

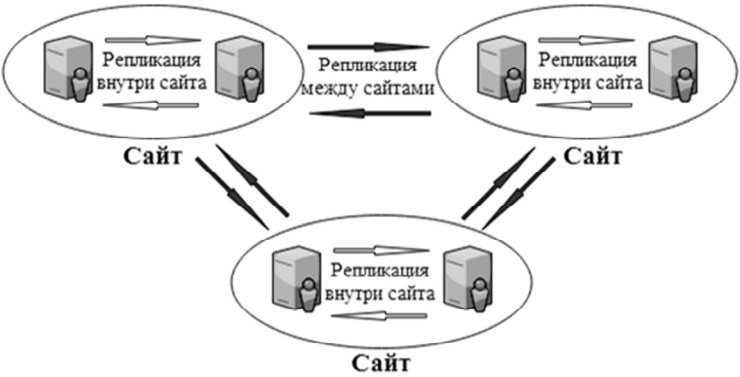


Рис. 4.3. Физическая структура Active Directory

Так как логическая и физическая структуры служат для решения разных задач, они практически не связаны между собой: общим объектом для них является контроллер домена, хранящий файл каталога *Ntds.dit*, в котором содержится информация об обеих структурах.

4.4. Практическое задание

Необходимо удовлетворить требования заказчика – внедрить в нашу распределенную корпоративную сеть сервер 1С-предприятие, чтобы каждый пользователь мог работать в этой программе без привязки к своему рабочему месту. На аналогичных условиях всем пользователям придется взаимодействовать с другими программами бух-

галтерии и своевременно сдавать отчеты как в электронном, так и в печатном виде.

После ознакомления с теоретическим материалом необходимо провести сравнительный анализ существующих технологий и продуктов; разработать и реализовать проект внедрения выбранного решения в тестовую корпоративную вычислительную сеть. В качестве примера стоит использовать внедрение программы 1С-предприятие и любых других сопутствующих бухгалтерских приложений, рассмотреть вопрос печати с удаленных объектов, усовершенствовать эшелон защиты путем внедрения системы управления событиями и сведениями о безопасности, провести итоговое тестирование реализованного комплекса защиты ИКТ. Дополнительным плюсом будет использование систем обнаружения и предотвращения вторжений.

Содержание отчета

1. Цель работы.
2. Техническое задание.
3. Сравнительный анализ существующих технологий.
4. Сравнительный анализ существующих решений.
5. Реализация проекта и комплекса мер по обеспечению ИБ.
6. Тестирование реализованного комплекса защиты ИКТ.
7. Заключение.
8. Список литературы.

Контрольные вопросы

1. Для каких целей служит продукт 1С-предприятие? В каких режимах он способен функционировать?
2. Перечислите механизмы обеспечения безопасности данных, предусмотренные в 1С-предприятии.
3. Что собой представляет сервер терминалов? Какими преимуществами и недостатками он обладает?
4. Изложите концептуальные основы технологии виртуализации приложений.
5. Какими возможностями обладает платформа *VMware Horizon*?
6. Для чего применяется служба каталогов *Active Directory*?
7. Как *Active Directory* связана с доменной системой имен?

8. Дайте определения понятию объекта, атрибута в терминах *AD*. Как они взаимосвязаны?

9. Для каких целей служит построение логической и физической структур *AD*?

10. Приведите основные компоненты каждой из них. Как взаимосвязаны данные структуры?

5. АУДИТ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ВЫЧИСЛИТЕЛЬНЫХ СЕТЕЙ И ИНФОРМАЦИОННЫХ СИСТЕМ

Ключевые слова: механизмы активного и пассивного анализа информационных систем и вычислительных сетей (*сетевые снифферы, сканеры, зондеры*)*, Kali Linux, Tails.*

Настоящий раздел посвящен изучению алгоритмов, методов и инструментов тестирования информационной безопасности вычислительных сетей и информационных систем, а также приобретению навыков проведения аудита информационной безопасности серверных решений.

5.1. Теоретические основы анализа уязвимостей информационных систем и вычислительных сетей

При выполнении тестирования безопасности вычислительных сетей и информационных систем невозможно обойтись без сетевых анализаторов трафика (снифферов). Под этим термином понимают программу или программно-аппаратное устройство, предназначенное для перехвата и последующего анализа либо только анализа сетевого трафика. Для корректной работы сниффера необходимо настроить сетевой интерфейс в режим «неразборчивого захвата» с целью обработки всего трафика сегмента сети.

Другим немаловажным инструментом выступают сканеры уязвимостей – это программные или аппаратно-программные средства, служащие для осуществления диагностики и мониторинга сетевых хостов, позволяющие сканировать сети, компьютеры и приложения на предмет обнаружения возможных проблем в системе безопасности, оценивать и устранять уязвимости. При этом стоит отметить, что большинство программных продуктов лишь выдает инструкции по устранению уязвимостей или их использованию.

Сканеры выявляют следующие типичные уязвимости:

1) неправильная настройка межсетевых экранов, сетевого оборудования, веб-серверов и баз данных;

2) ранее опубликованные и известные инструменты скрытного управления компьютером (бэкдоры);

3) восприимчивость к проникновению из незащищенных систем;

4) троянское программное обеспечение;

5) слабые пароли.

Функционировать такие средства могут на сетевом уровне (англ. *network-based*), уровне ОС (англ. *host-based*) и уровне приложения (англ. *application-based*). Наибольшее распространение получили средства анализа защищенности сетевых сервисов и протоколов. Другими по распространенности являются средства анализа защищенности ОС.

Помимо обнаружения уязвимостей при помощи средств анализа защищенности можно быстро определить все узлы корпоративной сети, доступные в момент проведения тестирования, выявить все используемые в ней сервисы и протоколы, их настройки и возможности для несанкционированного воздействия (как изнутри корпоративной сети, так и снаружи).

Существует два основных механизма, при помощи которых анализатор проверяет наличие уязвимости – сканирование и зондирование.

Сканирование – механизм пассивного анализа, с помощью которого сканер пытается определить наличие уязвимости без фактического подтверждения ее наличия по косвенным признакам. Этот метод является наиболее быстрым и простым для реализации.

Зондирование – механизм активного анализа, который позволяет убедиться, присутствует ли на анализируемом узле уязвимость. Зондирование выполняется путем имитации атаки, использующей проверяемую уязвимость.

5.2. Инструменты и средства анализа уязвимостей

Эти механизмы довольно часто реализуются следующими методами [46–51]. В первую очередь используется проверка заголовков (англ. *banner check*). Указанный механизм представляет собой ряд проверок типа сканирование и позволяет делать вывод об уязвимости, опираясь на информацию в заголовке ответа на запрос сканера. Типичный пример такой проверки – анализ заголовков программы *Sendmail* или *FTP*-сервера, позволяющий узнать их версию и на основе этой информации

сделать вывод о наличии в них уязвимости. Описываемый способ наиболее быстрый и простой для реализации проверки присутствия на сканируемом узле уязвимости.

Во вторую очередь задействуются активные зондирующие проверки (англ. *active probing check*), которые также относятся к механизму сканирования. Однако они основаны не на проверках версий программного обеспечения в заголовках, а на сравнении цифрового слепка (англ. *fingerprint*) фрагмента программного обеспечения со слепком известной уязвимости. Аналогичным образом поступают антивирусные системы, сравнивающие фрагменты сканируемого программного обеспечения с сигнатурами вирусов, хранящимися в специализированной базе данных. Описанный метод имеет хорошую скорость работы, но реализуется труднее, чем проверка заголовков.

И в заключение применяется имитация атак (англ. *exploit check*). Данные проверки относятся к механизму «зондирования» и основаны на эксплуатации различных дефектов в программном обеспечении. Частным случаем зондирования является пентестинг (англ. *pentest, penetration testing* – тестирование на проникновение).

Большинство сканеров проводят анализ защищенности в несколько этапов:

1) сбор информации о сети. На этом этапе идентифицируются все активные устройства в сети и определяются запущенные на них сервисы и домены;

2) обнаружение потенциальных уязвимостей. Сканер использует некую базу данных для сравнения собранной информации с известными уязвимостями при помощи проверки заголовков или активных зондирующих проверок;

3) подтверждение выбранных уязвимостей. Сканер использует специальные методы и моделирует/имитирует определенные атаки для подтверждения факта наличия уязвимостей на выбранных узлах сети;

4) генерация отчетов. На основе собранной информации система анализа защищенности создает отчеты, описывающие обнаруженные уязвимости;

5) автоматическое устранение уязвимостей или выдача инструкций по их нейтрализации или использованию. Этот этап очень редко реализуется в сетевых сканерах, но широко применяется в системных сканерах.

Наиболее содержательными продуктами в этой области являются узкоспециализированные программные решения от хакеров, описыва-

ющие и эксплуатирующие ранее неизвестные и официально не опубликованные уязвимости. Однако найти их можно зачастую лишь с использованием оверлейных сетей. Соответственно необходимо продемонстрировать навыки поиска, обработки и анализа информации.

Для обеспечения надлежащего уровня информационной безопасности любой информационный ресурс требует сопровождения квалифицированным техническим специалистом с целью устранения различных проблем, проведения профилактических работ, обновления программного обеспечения, изучения и устранения уязвимостей нулевого дня.

Одной из наиболее известных операционных систем в области информационной безопасности является *Kali Linux*. Этот дистрибутив, пришедший на замену ОС для проведения тестирования безопасности *BlackTrack*, имеет в своей основе *Debian GNU/Linux. Kali Linux* является специализированным инструментом и не предназначен для использования в качестве основного дистрибутива: в нем отсутствует большинство служб и программ, необходимых для решения повседневных задач.

Kali Linux содержит следующие категории программ:

1) сетевые сканеры – программы, определяющие конфигурации систем, открытые порты и работающие на них сервисы, а также выявляющие известные уязвимости;

2) снифферы, перехватывающие сетевой трафик;

3) эксплоит-базы – базы данных, включающие в себя большое количество готовых программ для использования известных уязвимостей ПО и ОС;

4) программы для подбора паролей (брутфорса);

5) прокси-инструментарий (более подробно данные инструменты рассматриваются в разделе настоящего пособия;

6) *Cisco*-инструментарий;

7) утилиты для работы в беспроводных сетях;

8) веб-инструментарий;

9) инструменты для работы с базами данных и т. д.

Перечислим наиболее известные:

- *Jhon The Ripper (JTR, Jhon)* – программа с открытым исходным кодом, позволяющая осуществлять подбор паролей (англ. *brute force*). Принцип работы *JTR* заключается в шифровании текстовых строк из словаря аналогично шифрованию пароля и сравнении с ним полученного результата;

- *Aircrack-ng* представляет собой набор инструментов для тестирования безопасности *wi-fi* сетей, дающий возможность выполнения мониторинга сетевого трафика, перебора ключей *WPA-PSK* и т. д.;
- *THC Hydra* – инструмент, работающий аналогично *JTR*, но *online*;
- *Burp Suite* – программа для поиска уявимостей веб-сайтов и приложений;
- *Wireshark* – сетевой анализатор трафика;
- *Nmap* – программа для тестирования безопасности сетей и сканирования портов;
- *Maltego* – инструмент аналитики, позволяющий находить и визуализировать связи между событиями и объектами;
- *Metasploit* – набор инструментов для тестирования известных уязвимостей;
- *Acunetix* – программа для сканирования веб-сайтов, позволяющая обнаружить вероятные *SQL*-инъекции, *XSS*, *CSRF* и т. д.;
- *Social-Engineer toolkit* – инструмент тестирования атак социального инжиниринга.

Стоит отметить, что все рассмотренные программы и утилиты предназначены исключительно для использования в благих целях. Настоящий раздел ориентирован на повышение квалификации в области информационной безопасности и не является пособием по проведению несанкционированного проникновения.

5.3. Практическое задание

Прежде всего необходимо овладеть теоретической частью материала и изучить алгоритмы, методы и инструменты тестирования информационной безопасности вычислительных сетей и информационных систем. Затем выполнить поиск и сравнительный анализ существующих решений (как бесплатных, так и проприетарных). Далее произвести аудит информационной безопасности ранее развернутого серверного решения с использованием рассмотренных инструментов. Проанализировать и устранить выявленные уязвимости объекта.

Содержание отчета

1. Цель работы.
2. Техническое задание.
3. Сравнительный анализ существующих решений.
4. Тестирование серверного решения.
5. Реализация комплекса мер по обеспечению ИБ.
6. Заключение.
7. Список литературы.

Контрольные вопросы

1. Что представляют собой сетевые анализаторы трафика?
2. Что представляют собой сканеры уязвимостей? Какие стандартные уязвимости они могут выявлять?
3. Помимо уязвимостей, какую информацию могут определить сканеры?
4. На каких уровнях способны функционировать сканеры уязвимостей? В чем состоят различия?
5. Перечислите механизмы выявления уязвимостей, используемые сканерами. Каковы методы их реализации?
6. Приведите пример поэтапного анализа защищенности информационной системы или сети.
7. Приведите примеры программных продуктов для выполнения анализа уязвимостей (бесплатных и проприетарных). Почему для их поиска рекомендуется использовать оверлейные сети?
8. Почему не рекомендуется использовать *Kali Linux* для повседневных задач?
9. Какие виды программ содержатся в дистрибутиве *Kali Linux*?
10. Перечислите дистрибутивы для проведения тестирования информационной безопасности, помимо *Kali Linux*. В чем состоит их различие?

6. СИСТЕМЫ ОБНАРУЖЕНИЯ И ПРЕДОТВРАЩЕНИЯ ВТОРЖЕНИЙ (*IDS/IPS*)

Ключевые слова: системы обнаружения и предотвращения вторжений (СОВ/СПВ или IDS/IPS), Snort, Suricata, Bro, сканеры, снифферы, зондеры.

Раздел посвящен изучению систем обнаружения и предотвращения вторжений, а также систем сбора и корреляции событий. Рассматривается архитектура, функции, алгоритмы и методы работы данных систем.

6.1. Архитектура и функции *IDS/IPS*

Системы обнаружения вторжений (англ. *Intrusion Detection System*, *IDS*) являются программными или аппаратно-программными системами, которые автоматизируют процесс просмотра событий, возникающих в компьютерной системе или сети, и анализируют их с точки зрения безопасности [52–56]. Такое определение можно считать корректным в рамках теории вычислительных машин, систем и сетей.

IDS состоят из трех функциональных компонентов: информационных источников, анализа и ответа. Система получает информацию о событии из одного или более источников информации, выполняет определяемый конфигурацией анализ данных события и затем создает специальные ответы – от простейших отчетов до активного вмешательства при определении проникновений.

Обнаружение проникновения является процессом мониторинга и анализа событий, происходящих в компьютерной системе или сети.

Проникновения определяются как попытки компрометации конфиденциальности, целостности, доступности или обхода механизмов безопасности компьютера или сети. Они могут осуществляться как атакующими, получающими доступ к системам из Интернета, так и авторизованными пользователями систем, пытающимися получить дополнительные привилегии, которых у них нет.

Наиболее распространенные функции системы обнаружения вторжений:

1) упреждающие воздействие с предупреждением об ответственности;

2) фильтрация информационных потоков;

3) идентификация преамбул атак (сетевого зондирования или некоторого другого тестирования для обнаружения уязвимостей и предотвращения их дальнейшего развития);

4) документирование существующих угроз для сети и систем;

5) обеспечение контроля качества разработки и администрирования;

6) получение полезной информации о проникновениях, которые имели место, с предоставлением улучшенной диагностики для восстановления и корректирования вызвавших проникновение факторов;

7) идентификация стороны вторжения.

Системой предотвращения вторжений (англ. *Intrusion Prevention System*, *IPS*) именуют *IDS* с инструментами автоматической защиты. Помимо пакетной обработки в современных продуктах часто встречается режим работы в «реальном времени» (англ. *Real-Time*).

6.2. Виды *IDS/IPS*-систем

Системы обнаружения и предотвращения вторжений могут функционировать на уровне сети, хоста и приложения.

Системы уровня сети (англ. *Network-based IDS/IPS*) состоят из множества информационных источников (датчиков), расположенных в ключевых точках сети (рис. 6.1). Они обладают рядом преимуществ:

1) грамотное расположение нескольких *NIDS/NIPS* позволяет просматривать большую сеть. Датчики сбора информации *NIDS/NIPS* должны быть установлены в каждом сегменте сети (например, посредством технологии зеркалирования на управляемых коммутаторах *L2+*);

2) *NIDS/NIPS* не оказывают влияния на нормальное функционирование сети, поэтому для их размещения не требуется модицифировать ее топологию;

3) *NIDS/NIPS* могут быть размещены невидимым злоумышленнику образом.

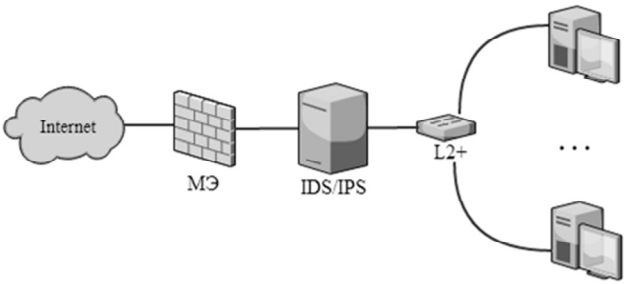


Рис. 6.1. IDS/IPS уровня сети

Однако подобные системы имеют и ряд недостатков:

1) распознавание атаки может быть затруднено при больших объемах трафика в крупных или занятых сетях;

2) отсутствие возможности анализа информации, передающейся в зашифрованном виде (если отсутствуют сертификаты и ключи для процедуры дешифрования);

3) у *NIDS/NIPS* вызывает затруднение определение сетевых атак, включающих фрагментированные пакеты.

IDS/IPS-системы уровня узла (англ. *Host-based IDS, HIDS/HIPS*) анализируют записи системного журнала и результаты аудита ОС, отслеживая атаки, направленные на конкретный хост (рис. 6.2).

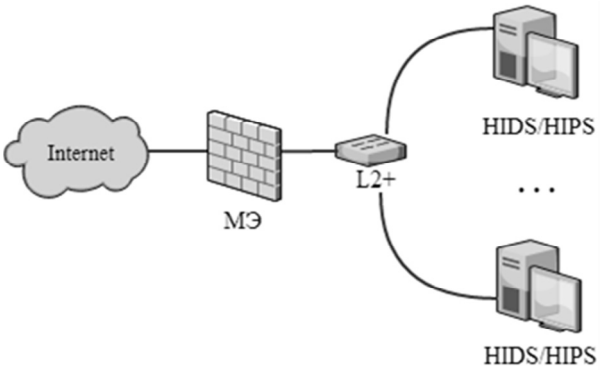


Рис. 6.2. IDS/IPS-системы уровня узла

По сравнению с сетевыми системам обнаружения и предотвращения вторжений *HIDS/HIPS* имеют следующие преимущества:

1) способны определять атаки, которые не видят *NIDS/NIPS,* поскольку анализ событий безопасности локально сосредоточен на одном узле;

2) способны работать в условиях зашифрованного сетевого трафика;

3) работа не зависит от наличия в сети коммутаторов и прочих сетевых устройств в отличие от *NIDS/NIPS*.

Недостатки *HIDS/HIPS*:

1) конфигурируются для каждого узла отдельно, соответственно данные системы более трудны в настройке и управлении;

2) вследствие расположения на потенциально атакуемом хосте сами *HIDS/HIPS* также могут подвергаться атакам;

3) не обладают возможностью обнаружения сканирования всей сети, поскольку анализируют пакеты, получаемые конкретным хостом;

4) могут быть блокированы некоторыми видами *DoS*-атак;

5) расходуют вычислительные ресурсы наблюдаемых хостов;

6) из-за использования большого объема данных (результаты аудита ОС) *HIDS/HIPS* может потребоваться дополнительное локальное хранение информации.

IDS/IPS уровня приложения (англ. *Application-based IDS/IPS*) в общем случае можно назвать подмножеством *HIDS/HIPS.* Они взаимодействуют непосредственно с приложением и анализируют события, поступившие в его ПО.

Преимущества данных систем:

1) способны отслеживать неавторизованную деятельность пользователей посредством анализа их взаимодействия с приложениями;

2) способны работать в окружениях с зашифрованным трафиком.

Среди их недостатков выделяют:

1) могут быть уязвимы в случае атаки на логи приложений, поскольку они зачастую защищены не столь надежно, как результаты аудита ОС, используемые *IDS*-системами уровня узла;

2) не способны определять атаки, связанные с нарушением целостности ПО (троянские программы и т. д.).

6.3. Подходы к анализу событий безопасности

Стоит выделить основные методы работы систем обнаружения и предотвращения вторжений: сигнатурный, поведенческий, идентификации аномалий и комбинированный.

Сигнатурный метод использует анализ соответствия событий безопасности системы заданному образцу, описывающему конкретную атаку. Этот способ показывает высокую эффективность при распознавании известных атак и средств их реализации, к тому же он не зависит от квалификации системного администратора. Однако существенным недостатком определения злоупотреблений является отсутствие возможности определения атак, чьи сигнатуры отсутствуют в их базе данных, а также сложных комбинированных атак.

Метод идентификации аномалий представляет собой определение необычного поведения пользователя и системы. Исходя из данных истории создаются профили, описывающие стандартную деятельность пользователей или узлов, и далее с использованием ряда метрик – статистических, нейронных сетей, генетических алгоритмов – определя-

ется отклонение анализируемого поведения от нормального. Определение аномалий, в отличие от сигнатурного метода, позволяет идентифицировать ранее неизвестные атаки без знания их деталей, однако данный метод создает большое число ложных срабатываний при непредсказуемой сетевой активности. Помимо этого он требует временных затрат на обучение системы защиты для определения показателей нормального поведения. Синонимом метода идентификации аномалий часто считают метод поведенческого анализа, хотя в научной литературе эти понятия принято различать.

В дополнение к описанным методам обнаружения вторжений следует назвать собственные политики безопасности системы. Описание правил сетевой безопасности требует высокой профессиональной подготовки от администратора и является более трудоемким процессом, чем использование стандартных методов анализа событий системы. Однако подобный подход отличается наибольшей гибкостью и дает возможность распознавания новых атак. Комбинирование сигнатурного анализа, метода аномалий и применение политик безопасности позволит значительно повысить уровень защищенности сети. Стоит отметить, что в реальных системах, где используются криптографические протоколы и оверлейные сети, такие методы не оправдывают ожиданий.

Для обеспечения комплексной защиты информационных ресурсов нельзя забывать о системах управления событиями и сведениями о безопасности, часто именуемых системами сбора и корреляции событий (англ. *Security Information and Event Management, SIEM*). Функциональное назначение данного инструмента – анализ информации, поступающей от различных систем (ОС, антивирусов, программ сканирования и фильтрации, *DLP*, *IDS*/*IPS*, маршрутизаторов, межсетевых экранов и других приложений/сетевых узлов), а также детектирование отклонений от норм различных параметров. При обнаружении отклонения система генерирует инцидент. Статистические и математические технологии являются неотъемлемой частью работы *SIEM*. Данные системы позволяют контролировать требования нормативных документов и политик безопасности, своевременно выявлять атаки и оперативно на них реагировать.

Помимо этого *SIEM* способны разрешить ряд типовых проблем, с которыми может столкнуться администратор сети:

1) осуществляют сбор событий с практически любых информационных источников и приводят их к единообразному виду, что упрощает задачу анализа событий безопасности;

2) способны агрегировать однотипные события, тем самым они уменьшают объем данных для анализа без потери потенциально важных инцидентов;

3) позволяют выявлять сложные распределенные атаки на основе сопоставления событий, которые на первый взгляд могут казаться несвязанными;

4) решают задачу централизованного хранения событий различных систем, используя сжатие.

Важным аспектом внедрения в корпоративную сеть *SIEM* являются временные затраты на сбор статистики о стандартном поведении узлов. Тестовая эксплуатация данных систем может длиться в течение нескольких месяцев, причем большинство срабатываний в этот период будут ложнопозитивными. Таким образом, требуется четко осознавать, что установка системы сбора и корреляции событий для усиления эшелона защиты сети является необходимой в случае крупной распределенной инфраструктуры сети, наличия в ней большого количества устройств обеспечения сетевой информационной безопасности (межсетевых экранов, систем обнаружения и предотвращения вторжений, *DLP*-систем, антивирусов и т. д.), присутствия в инфраструктуре сети специфических устройств, требований на соответствие различным стандартам, а также частой необходимости расследования инцидентов (к примеру, при выявлении более ста попыток проникновения извне в день).

6.4. Практическое задание

После ознакомления с теоретическим материалом необходимо провести сравнительный анализ существующих систем обнаружения и предотвращения вторжений. Разработать и реализовать проект внедрения выбранного продукта в тестовую корпоративную вычислительную сеть. Исследовать алгоритмы и методы работы данных систем посредством запуска механизмов активного и пассивного анализа объектов (сканеров, снифферов, зондеров). Усовершенствовать эшелон защиты путем внедрения системы управления событиями и сведениями о безопасности. Провести итоговое тестирование реализованного комплекса защиты ИКТ.

Содержание отчета

1. Цель работы.
2. Техническое задание.
3. Сравнительный анализ существующих решений.
4. Реализация комплекса мер по обеспечению ИБ.
5. Тестирование реализованного комплекса защиты ИКТ.
6. Заключение.
7. Список литературы.

Контрольные вопросы

1. Что называют системой обнаружения и предотвращения вторжений (*IDS/IPS*)?
2. Какие функции включают данные системы?
3. Опишите архитектуру *IDS/IPS*-систем.
4. На каких уровнях способны функционировать *IDS/IPS*? В чем состоит их различие? Каковы их преимущества и недостатки?
5. В чем состоит отличие данного класса систем от межсетевых экранов?
6. Перечислите методы анализа событий безопасности в системах обнаружения вторжений.
7. Какой из данных методов является наилучшим? Ответ обосновать.
8. Что называют системой сбора и корреляцией событий (*SIEM*)?
9. Приведите основные функции *SIEM*-систем.
10. Способна ли данная система заменить *IDS/IPS*? Ответ обосновать.

7. ПРОГРАММНЫЕ КРИПТОГРАФИЧЕСКИЕ МЕХАНИЗМЫ ЗАЩИТЫ ИНФОРМАЦИИ

Ключевые слова: симметричное и асимметричное шифрование, PGP, OTR, GPG4WIN, XMPP, Tox, TrueCrypt, AxCrypt, DiskCryptor, FreeOTFE, BestCrypt.

Настоящий раздел посвящен изучению криптографических средств защиты информации. Приводятся базовые теоретические сведения о криптографии и криптографических алгоритмах.

Специалистам информационной безопасности требуется работать с юридическими документами как Российской Федерации, так и с международными. Для приобретения опыта работы с государственной терминологией в данной теоретической части представлены выдержки по ключевым определениям рассматриваемой области.

Криптография – дисциплина, включающая в себя принципы, средства и методы преобразования информации в целях сокрытия ее содержания, предотвращения не поддающегося обнаружению ее видоизменения или несанкционированного использования. Криптография ограничена преобразованием информации с использованием одного или более секретных параметров (например, криптографических переменных) или соответствующим управлением ключом.

Криптографическая защита информации (зашифрование, расшифрование) – процесс преобразования открытой информации с целью сохранения ее в тайне от посторонних лиц при помощи некоторого алгоритма, называемого шифром. Для зашифрования информации сторонами используется алгоритм криптографического преобразования, реализованный в сертифицированном средстве криптографической защиты информации. Расшифрование является обратным процессом зашифрования.

Средства шифрования – аппаратные, программные и программно-аппаратные шифровальные (криптографические) средства, реализующие алгоритмы криптографического преобразования информации для ограничения доступа к ней, в том числе при ее хранении, обработке и передаче.

Средства криптографической защиты информации (СКЗИ) – сертифицированные в порядке, установленном законодательством Российской Федерации, аппаратные и (или) программные средства, обеспечивающие шифрование, контроль целостности и применение ЭП при обмене электронными документами.

Электронная цифровая подпись (ЭЦП) – последовательность символов, полученная в результате криптографического преобразования исходной информации, позволяющей подтверждать целостность и неизменность этой информации, а также ее авторство.

Под технической защитой конфиденциальной информации понимается выполнение работ и (или) оказание услуг по ее защите от несанкционированного доступа, от утечки по техническим каналам, а также от специальных воздействий на такую информацию в целях ее уничтожения, искажения или блокирования доступа к ней.

Аутентификация – проверка принадлежности субъекту доступа предъявленного им идентификатора (подтверждение подлинности).

Симметричный алгоритм – криптографический алгоритм, использующий один и тот же ключ как для шифрования, так и для дешифрования. При использовании симметричного алгоритма информационный обмен имеет следующий вид:

1) отправитель передает ключ получателю;

2) с помощью ключа отправитель зашифровывает пересылаемое получателю сообщение;

3) после прихода сообщения получателю он его расшифровывает.

Асимметричный алгоритм – криптографический алгоритм, использующий различные математически связанные ключи для шифрования и дешифрования. Этапы обмена информацией с использованием данного алгоритма выглядят следующим образом:

1) получатель вычисляет открытый и секретный ключ; секретный ключ хранится втайне, тогда как открытый является общедоступным;

2) с помощью открытого ключа получателя отправитель зашифровывает пересылаемое получателю сообщение;

3) сообщение приходит получателю и расшифровывается им с помощью секретного ключа.

К сожалению, законодательство РФ в области информационных технологий является недоработанным. Даже в приведенных определениях есть ряд неточностей, исправив которые можно улучшить свой итоговый балл за выполнение задания.

Рассмотрим два подхода к обеспечению криптографической защиты информации, которые необходимо применить в ходе выполнения практического задания к данному разделу, – локальный и удаленный [57].

7.1. Локальный подход к обеспечению криптографической защиты информации

Важным аспектом, требующим внимания, является потенциальная возможность перехвата данных и кражи оборудования. К сожалению, установка паролей (*BIOS*, ОС, архивы, офисные документы), использование штатного шифрования не обеспечивают надлежащего уровня безопасности информационных ресурсов и могут быть упразднены техническим специалистом в считанные минуты. Соответственно

необходимо применить криптографические средства для защиты системного раздела/диска.

Наиболее известным инструментом шифрования «на лету» (англ. *on-the-fly* – автоматическое шифрование или дешифрование данных во время их считывания/записи) является *TrueCrypt*.

Программа *TrueCrypt* обладает широким спектром возможностей:

1) создание зашифрованного виртуального диска тремя различными способами: в файле-контейнере, что значительно облегчает работу с ним (перенос, копирование, переименование и т. д.); в виде зашифрованного раздела диска; посредством полного шифрования содержимого устройства (например, *USB* флеш-память);

2) обеспечение двух уровней правдоподобного отрицания наличия зашифрованных данных: возможно создание скрытого тома и задание второго пароля для обычного тома с целью получения доступа к данным, недоступным по основному паролю;

3) отсутствие возможности идентификации томов *TrueCrypt*: тома *TrueCrypt* неотличимы от набора случайных данных, поэтому созданные программой файлы нельзя связать с ней;

4) поддержка симметричных алгоритмов блочного шифрования *AES*, *Serpent* и *Twofish*;

5) возможность выбора одной из трех хеш-функций: *HMAC-RIPEMD-160*, *HMAC-Whirlpool* и *HMAC-SHA-512* для генерации ключей шифрования, модификатора и ключа заголовка;

6) переносимость – *TrueCrypt* может запускаться без установки в ОС;

7) возможность изменения паролей и ключевых фраз для доступа к томам без потери зашифрованных данных;

8) поддержка создания зашифрованного динамического файла на дисках *NTFS* (англ. *New Technology File System*);

9) возможность резервного сохранения и восстановления заголовков томов (к примеру, при необходимости монтирования тома после аппаратной ошибки, приведшей к повреждению его заголовка).

В настоящее время данный программный продукт перестал поддерживаться разработчиками. Наиболее доверенной версией является *TrueCrypt 7.1*.

Помимо *TrueCrypt* на рынке представлено множество программ со схожими функциями: *BitLocker*, *AxCrypt*, *DiskCryptor*, *BestCrypt, FreeOTFE* и т. д. В ходе выполнения задания необходимо будет выполнить сравнительный анализ этих инструментов и произвести аргументированный выбор наиболее подходящего из них.

7.2. Методы и средства обеспечения информационной безопасности удаленного взаимодействия

Под удаленным подходом к обеспечению криптографической защиты информации подразумевается обеспечение информационной безопасности удаленной коммуникации с помощью технологий шифрования. В качестве примера можно привести расширяемый протокол обмена сообщениями и информацией о присутствии *XMPP* (англ. *Extensible Messaging and Presence Protocol*). В основе данного протокола лежит расширяемый язык разметки *XML* (англ. *eXtensible Markup Language)*. Архитектура *XMPP* сходна с другими протоколами прикладного уровня: каждый клиент имеет уникальное имя и выполняет информационный обмен с другими клиентами через сервер. Обмен данными между клиентом и сервером инкапсулируется в *XML*-потоки. Другим допустимым решением для организации мгновенного обмена сообщениями может выступать протокол *Tox*. Для обеспечения взаимодействия клиентов используется пиринговый обмен данными. После инсталляции *Tox* создается пара ключей: публичный, служащий в качестве уникального идентификатора собеседника, и секретный, хранящийся исключительно у владельца и подтверждающий его подлинность. Поиск собеседников выполняется с помощью распределенной хэш-таблицы. Защита соединения обеспечивается использованием *SOCKS*-прокси серверами; помимо этого существует возможность перенаправления трафика через оверлейную сеть *Tor* (подробнее рассматривается в разделе 9 настоящего пособия.

Важно понимать, что использование данных протоколов в стандартном виде не гарантирует достаточного уровня безопасности передаваемых данных, поэтому рекомендуется применение дополнительных криптографических средств.

Возможным является использование криптографического протокола для систем мгновенного обмена сообщениями *OTR* (англ. O*ff-the-Record*). Этот протокол удовлетворяет трем основным требованиям: шифрование передаваемых данных, аутентификация сторон и отсутствие возможности компрометации переписки в случае утери ключей. Последнее достигается за счет постоянного обновления ключей во время обмена сообщениями. Для установления общего секретного ключа применяется протокол Диффи–Хеллмана (англ. *Diffie–Hellman*, *DH*), дающий возможность двум и более сторонам получить общий

секретный ключ с использованием незащищенного канала связи и в дальнейшем применять его для шифрования информационного обмена. Шифрование в *OTR* обеспечивается с помощью алгоритма *AES* (англ. *Advanced Encryption Standard*). Для аутентификации сообщений применяется ключ, полученный хешированием ключа, который используется для шифрования сообщения.

Этот протокол предназначен для использования только двумя сторонами. В случае необходимости одновременного использования протокола несколькими пользователями возможна установка расширений *OTR – GOTR* (*GroupOTR*) и *mpOTR* (*Multy-Party OTR*).

Не стоит забывать и о криптографической системе *PGP* (англ. *Pretty Good Privacy*), позволяющей производить операции шифрования и цифровой подписи сообщений, файлов и прочих данных, представленных в электронном виде, включая прозрачное шифрование информации на запоминающих устройствах, к примеру, на жестком диске.

7.3. Практическое задание

Требования заказчика – подготовить ноутбук и смартфон директора перед поездкой в командировку, а также проработать вопрос его безопасной коммуникации как с персоналом, так и со сторонними оппонентами. По имеющимся данным, конкуренты предпримут попытки по перехвату конфиденциальной информации.

В первую очередь необходимо произвести конфигурирование объектов защиты: ноутбука и смартфона. Стоит запретить все виды удаленного доступа, настроить брандмауэр, устранить уязвимости и бэкдоры, а также установить дополнительные средства защиты (*IDS*/*IPS*, *AV* и т. д.).

Далее необходимо применить криптографические средства для защиты системного раздела/диска. При принятии любого решения необходимо будет провести сравнительный анализ существующих продуктов.

Для защиты процесса передачи данных также неизбежно использование технологии шифрования. Выбираемые алгоритмы, методы и протоколы должны обладать высокой степенью криптоустойчивости. Необходимо проработать два вопроса: мгновенный обмен сообщениями между сотрудниками посредством корпоративного сервера и с оппонентами через сторонние сервисы и программное обеспечение (ПО).

Свободное программное обеспечение с открытым исходным кодом является более доверительным в сравнении с проприетарными закрытыми решениями.

Содержание отчета

1. Цель работы.
2. Техническое задание.
3. Сравнительный анализ существующих решений.
4. Реализация комплекса мер по обеспечению ИБ.
5. Заключение.
6. Список литературы.

Контрольные вопросы

1. Что такое криптография? Для чего необходима криптографическая защита информации?
2. Дайте определение электронной цифровой подписи (ЭЦП). Для чего она применяется?
3. В чем состоят симметричный и асимметричный алгоритмы шифрования данных?
4. Объясните необходимость применения дополнительных средств шифрования системных разделов/дисков.
5. Перечислите возможности обеспечения криптографической защиты информации, реализованные в программном продукте *TrueCrypt*.
6. Что подразумевается под обеспечением двух уровней правдоподобного отрицания наличия зашифрованных данных?
7. Какие аналоги *TrueCrypt* вы знаете? В чем заключается их различие?
8. Приведите существенные различия протоколов *OTR* и *PGP*.
9. С помощью каких инструментов возможна организация мгновенного обмена сообщениями? Ответ обосновать.
10. Какие могут быть установлены дополнительные средства защиты ноутбука, смартфона?

8. БЕЗОПАСНОСТЬ ВИРТУАЛЬНЫХ ИНФРАСТРУКТУР

Ключевые слова: VMware ESXi, MS Hyper-V, OpenVZ, KVM, Xen.

Раздел посвящен изучению технологий виртуализации. Приводятся теоретические сведения о различных методах виртуализации и анализ существующих решений в данной области. Рассматриваются принципы работы, механизмы обработки запросов к аппаратным ресурсам и механизмы управления ими, а также отличительные особенности каждой системы.

8.1. Теоретические основы технологий виртуализации

Виртуализация – предоставление набора вычислительных ресурсов или их логического объединения, абстрагированное от аппаратного комплекса и обеспечивающее при этом логическую изоляцию вычислительных процессов, выполняемых на одном физическом ресурсе [58–59].

Паравиртуализация – технология виртуализации с применением модификации гостевых операционных систем. В свою очередь, гостевые объекты взаимодействуют с программой гипервизора, который предоставляет ей *API*, вместо прямого использования аппаратных ресурсов. Гипервизор осуществляет разделение и управление ресурсами, предоставляемыми гостевым ОС, а также взаимодействие запущенных ОС и их изоляцию друг от друга. Внесение изменений в гостевую ОС для гипервизора ранее было возможно лишь в случае наличия у нее открытого исходного кода, однако у современных систем подобного недостатка нет. Метод виртуализации позволяет достичь производительности, близкой к производительности реальной системы.

Аппаратная виртуализация реализуется с поддержкой специальной процессорной архитектуры. Это делает возможным использование изолированных гостевых систем, не зависящих от реализации платформы виртуализации и напрямую управляемых гипервизором. Применение аппаратной виртуализации обеспечивает производительность, сравнимую с производительностью невиртуализированной машины. Наиболее распространены технологии виртуализации *Intel-VT* (англ. *Intel Virtualization Technology*) и *AMD-V*.

Рассмотрим программные продукты, реализующие описанные технологии: *VMware ESXi, MS Hyper-V, OpenVZ, KVM.*

8.2. Программные средства виртуализации

Наиболее известным аппаратным гипервизором является *VMware ESXi,* поставляемый в качестве компонента *VMware vSphere.* Он устанавливается непосредственно на физический сервер, не требующий наличия установленной на нем ОС, и выполняет его разделение на несколько логических разделов (виртуальных машин). Преимуществами данного продукта являются простота развертывания, настройки и управления, а также небольшой размер. Управляют *VMware ESXi*

с помощью *API*-интерфейсов, что дает возможность проведения мониторинга оборудования и управления системой без устанавления агентов. В дополнение предоставляются интерфейсы удаленных командных строк – *vSphere Command Line Interface* (*vCLI*) и *PowerCLI* – для реализации настройки системы и устранения неполадок. Реализована поддержка нескольких методов развертывания: с использованием программы установки *ESXi*, с помощью сценариев и с применением среды *PXE (*англ. *Preboot eXecution Environment*). Для автоматизации повседневных задач в *ESXi* предусмотрен механизм создания сценариев. Кроме того, существует возможность подключения узлов *vSphere ESXi* к домену *Active Directory,* что избавляет от необходимости создания на каждом узле локальных учетных записей пользователей.

Microsoft Hyper-V является системой аппаратной виртуализации на основе гипервизора. Логической единицей разграничения, поддерживаемой данным гипервизором, является раздел. Каждый экземпляр гипервизора содержит один родительский раздел с запущенной ОС *Windows Server 2008*. На родительском разделе запускается стек виртуализации, имеющий прямой доступ к аппаратным устройствам; кроме того, он порождает дочерние разделы, где располагаются гостевые ОС. Дочерние разделы не обладают доступом к аппаратным ресурсам – они имеют виртуальное представление ресурсов, именуемое виртуальными устройствами. Все обращения к виртуальным устройствам выполняются через *VMBus* – логический канал, обеспечивающий взаимодействие между разделами, и перенаправляются к устройствам родительского раздела для получения доступа к физическим устройствам. Далее после запуска родительскими разделами *VSP* (англ. *Virtualization Service Provider* – провайдер сервиса виртуализации) и его соединения с *VMBus* выполняется обработка запросов дочерних разделов. Запросы к *VSP* родительского раздела перенаправляются через *VMBus* клиентом сервиса виртуализации *VSC* (англ. *Virtualization Service Client*). Реализована также поддержка виртуальными устройствами технологии прогрессивного ввода-вывода (англ. *Enlightened I/O*), позволяющая работать с *VMBus* напрямую, что делает возможным параллельную обработку любых уровней эмуляции устройства.

OpenVZ является реализацией технологии виртуализации на уровне ОС и базируется на ядре *Linux*, вследствие чего в качестве гостевых ОС могут выступать только дистрибутивы *Linux*. Такая технология позволяет запускать множество изолированных друг от друга копий ОС, называемых виртуальными средами (англ. *VE – Virtual Environ-*

ment) или виртуальными частными серверами (англ. *VPS – Virtual Private Servers*). Каждая виртуальная среда содержит собственные файлы, пользователей, группы, деревья процессов, сети, устройства и объекты *IPC* (англ. *inter-process communication* – межпроцессорное взаимодействие). Управление ресурсами в *OpenVZ* включает три компонента: двухуровневую дисковую квоту, честный планировщик процессора и набор счетчиков, ограничений и гарантий на каждую *VE*, именуемый «*User Beancounters»*. Во время работы виртуальных сред ресурсы могут быть изменены без требования перезагрузки. Отличительными особенностями *OpenVZ* являются масштабируемость, плотность и возможность массового управления *VE*.

KVM (англ. *Kernel-based Virtual Machine*) является инструментом виртуализации в среде *Linux* с поддержкой аппаратной виртуализации на базе *IntelVT* или *AMD SVM*. В программное обеспечение *KVM* входят загружаемый модуль ядра для предоставления основного сервиса виртуализации, процессорно-специфический загружаемый модуль *kvm-amd.ko* (*kvm-intel.ko*) и компоненты пользовательского режима. Основной концепцией, заложенной в *KVM*, является использование ядра *Linux* в качестве гипервизора. Устройство */dev/kvm* экспортируется модулем ядра и делает возможным его гостевой режим. Каждая виртуальная машина (ВМ) имеет собственное адресное пространство, отдельное от других ВМ и от адресного пространства ядра, а также собственное виртуальное программное обеспечение. Операции ввода/вывода с гостевой операционной системы производятся *QEMU*-платформой для эмуляции аппаратного обеспечения (дисков, графических адаптеров, сетевых устройств и т. д.). Любые запросы ввода/вывода гостевой ОС перенаправляются в пользовательский режим для эмулирования посредством процесса *QEMU*. Управление ресурсами в *KVM* осуществляется с помощью механизма ядра *Linux CGroups* (англ. *Control Group*).

8.3. Практическое задание

Требование заказчика – перевести серверный сегмент корпоративного ИКТ сектора на выделенные серверы дата-центра с использованием технологий виртуализации. Разработайте проект по миграции серверных решений и защите информационных ресурсов виртуальной инфраструктуры (на примере гипервизора/паравиртуализатора).

Необходимо провести детальный сравнительный анализ существующих решений в области виртуализации. Идентифицировать и устранить существующие уязвимости выбранного объекта. Спроектировать корпоративную вычислительную сеть (Интранет) с эшелонами защиты. Рекомендуется не упускать из виду обеспечение безопасности инструментов администрирования, резервного копирования, живой миграции хостов и балансировки нагрузки, а также настройку виртуальных коммутаторов и маршрутизаторов. При этом стоит проработать вопрос о целесообразности использования интеллектуальных функций сетевых узлов (*port security*, *IP-binding*, *ACL*, *VLAN*, *SNMP* и т. д.).

Содержание отчета

1. Цель работы.
2. Техническое задание.
3. Сравнительный анализ существующих решений.
4. Исследование выбранного гипервизора/паравиртуализатора.
5. Проектирование архитектуры и систем защиты.
6. Поиск и устранение уязвимостей.
7. Заключение.
8. Список литературы.

Контрольные вопросы

1. Что представляет собой программная виртуализация?
2. Что представляет собой аппаратная виртуализация?
3. Что представляет собой паравиртуализация?
4. Приведите принципиальные различия данных технологий. Каковы их преимущества и недостатки?
5. Для каких целей служит гипервизор?
6. Опишите принцип работы систем *ESXi*, *MS Hyper-V, OpenVZ, KVM*. В чем состоит их различие?
7. Какими уязвимостями обладают рассмотренные инструменты виртуализации?
8. Приведите сравнительный анализ данных систем с инструментами, не рассмотренными в настоящем разделе.

9. АНОНИМИЗАЦИЯ В ГЛОБАЛЬНОЙ СЕТИ ИНТЕРНЕТ

Ключевые слова: Proxy, технологии защищенного канала связи, VPN, Tor, I2P, TAILS, Kali Linux, «интернет-серфинг».

Настоящий раздел посвящен технологиям анонимизации и защиты процесса передачи данных в глобальной сети Интернет. Рассматриваются механизмы проксирования, а также различные реализации технологий виртуальных защищенных каналов связи и оверлейных сетей.

9.1. Механизмы проксирования

В случае если целью является экономия трафика вычислительной сети (посредством сжатия, кэширования и т. д.), ограничение доступа пользователей, то следует рассмотреть механизмы проксирования. Прокси-серверы принимают заявки от пользователей и выполняют запросы к различным сетевым службам от своего имени, что позволяет скрыть местоположение узла-отправителя.

Технология проксирования может применяться для различных целей:

1) экономия сетевого трафика посредством сжатия – данные из Интернета загружаются прокси-сервером и передаются пользователю в сжатом виде;

2) кэширование данных – в случае частых обращений к каким-либо внешним ресурсам разумно хранить их копию на прокси-сервере, выдавая ее по запросу. Таким образом, клиент быстрее получит запрашиваемую информацию, а нагрузка на канал связи во внешнюю сеть снизится;

3) ограничение доступа пользователей – ограничение на использование Интернета конкретным пользователям, запрет на доступ к каким-либо веб-сайтам, фильтрация рекламы, установление квоты на сетевой трафик и т. д.;

4) ограничение доступа к локальной сети извне – возможным вариантом настройки прокси-сервера является случай, когда локальные узлы обращаются к внешним только через него, тогда как внешние совсем не могут обращаться к локальным;

5) анонимизация – прокси-сервер способен скрывать или искажать информацию об узле-отправителе, тогда целевому серверу доступны лишь данные о самом прокси.

По принципу передачи данных выделяют две группы прокси-серверов: прозрачные и непрозрачные. Их различия обусловлены модификацией сообщений, передаваемых через прокси. Прозрачный прокси-сервер только при необходимости может вносить изменения в запрос или ответ, к примеру, добавить идентификационную информацию о себе или отправителе сообщения. При этом он обязан обеспечить неизменность длины содержимого передаваемого сообщения. Непрозрачный прокси-сервер способен модифицировать запрос и/или ответ: скрывать данные о клиенте, преобразовывать формат для уменьшения размера ответа, производить перевод текстового документа и т. д.

Как прозрачный, так и непрозрачный прокси-сервер могут иметь ассоциированный с ними кэш. Также оба вида прокси-серверов играют роль промежуточного звена между веб-клиентом и веб-сервером и осуществляют обмен сообщениями в формате *HTTP* (англ. *HyperText Transfer Protocol*) (рис. 9.1).

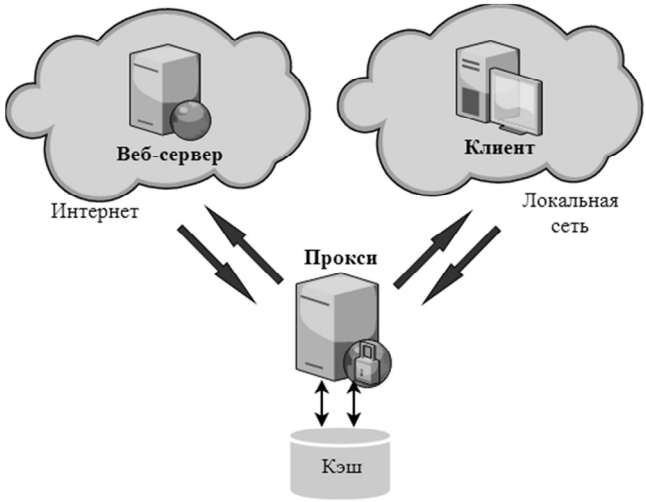


Рис. 9.1. Функционирование прокси-сервера

По выполняемому функционалу прокси-серверы разделяют на несколько видов:

1) *HTTP/HTTPS*-прокси являются наиболее распространенным типом прокси-серверов. В зависимости от уровня обеспечиваемой ано-

нимности они подразделяются на следующие категории: прозрачные – не скрывающие истинный *IP*-адрес клиента; анонимные – указывающие на использование прокси, но не показывающие истинный *IP*-адрес клиента; искажающие – модифицирующие *IP*-адрес клиента; элитные – не указывающие на использование прокси и скрывающие *IP*-адрес клиента;

2) *SOCKS*-прокси осуществляет передачу данных от клиента серверу без внесения в них изменений. Для веб-сервера *SOCKS*-прокси является клиентом. Наиболее актуальной версией протокола является *SOCKS 5*;

3) *FTP*-прокси предназначен для загрузки данных на файловые серверы;

4) *CGI*-прокси позволяют осуществлять анонимный переход с одной веб-страницы на другую. Данный тип прокси-серверов не требует изменения клиентских настроек.

В следующих разделах пособия описываются проекты оверлейных сетей, для реализации которых используются прокси-серверы.

9.2. Технологии виртуальных защищенных каналов связи

Рассмотрим технологии защищенного канала, задача которых заключается в обеспечении безопасности передачи данных по открытой транспортной сети. Наиболее распространенный вариант использования в глобальной сети Интернет, построенной на стеке протоколов *TCP/IP*. Таким образом, в сетях с коммутацией пакетов устанавливается виртуальная коммутация защищенных каналов, включающих в себя выполнение трех основных функций:

1) взаимная аутентификация абонентов;

2) защита передаваемых по каналу сообщений от несанкционированного доступа;

3) подтверждение целостности поступающих по каналу сообщений.

Важной характеристикой стандартов защищенного канала является уровень модели стека *TCP/IP*, на котором функционируют данные протоколы:

1) прикладной (англ. *Application layer*) – *S/MIME / PGP / HTTPS* и др.;

2) транспортный (англ. *Transport layer*) – *SSL / TLS / SOCKS* и др.;

3) сетевой (англ. *Internet layer) – IPSec* (*AH*, *ESP*) и др.;

4) канальный (англ. *Link layer*) – *PPTP / L2TP / PAP/ MS-CHAP* и др.

Виртуальной частной сетью (англ. *Virtual Private Network*) называют объединение защищенных каналов связи. Данные технологии позволяют разворачивать логическую сеть поверх уже существующей. В зависимости от уровня доверия к базовой среде виртуальные частные сети разделяют на защищенные, создаваемые внутри ненадежных сетей (*IPSec*, *OpenVPN* и *PPTP*), и доверительные, применяемые в случаях, когда базовая сеть считается безопасной (*L2P + IPSec*).

Наиболее известными реализациями технологий виртуальных частных сетей являются протоколы *IPsec* (англ. *IP Security*) и *OpenVPN*. Существует два способа функционирования *IPsec*: в туннельном и транспортном режимах. При работе в туннельном режиме шифрованию подвергается весь *IP*-пакет, тогда как в транспортном шифруется лишь его содержимое, причем соединение между узлами может быть образовано посредством других технологий (*L2P* и т. д.). Протокол *OpenVPN* для обеспечения безопасности передаваемых данных использует библиотеку *OpenSSL* и является менее криптоустойчивым решением по сравнению с *IPsec*, однако при наличии *NAT* он является предпочтительным вследствие более корректной работы.

Существует широкий спектр уязвимостей в алгоритмах данных протоколов (обусловленных «жесткой» логикой поведения) и их аппаратно-программной реализации. На всеобщий обзор уязвимости выносятся, как правило, минимум через год. Например, уязвимости *OpenSSL* "*Heartbeat*" и *MITM* были опубликованы на тематических ресурсах хакеров в скрытой оверлейной сети Интернета *I2P* в 2012 году, а выпуск официальных бюллетеней безопасности *CVE*-2014-0160 и *CVE*-2014-0224 состоялся лишь в апреле и июне 2014 года соответственно.

Стоит отметить, что при корректной настройке и дальнейшем сопровождении техническим специалистом данная технология обеспечивает надежный уровень информационной безопасности.

9.3. Проект *JAP*

Одним из ключевых трендов развития ИКТ является исследование и разработка оверлейных сетей (от англ. *Overlay Network*). Под данным термином подразумевается организация логической сети, функционирующей поверх существующей глобальной вычислительной сети Интернет.

Узкоспециализированные анонимные/анонимизирующие сети являются наиболее простым видом оверлейных сетей. В качестве примера рассмотрим проект *JAP* (англ. *Java Anonymous Proxy*). Его целевое назначение – анонимизация работы протокола передачи гипертекста *HTTP*, т. е. веб-трафика.

Используется метод управления трафиком на основе микс-узлов, представленный на рис. 9.2. Клиент отправляет данные не искомому адресату, а на хост каскадов микс-серверов, которые мультиплексируют информационные потоки различных клиентов и отправляют запросы их реальным адресатам. Ответы транслируются по тому же маршруту. Клиент-серверное взаимодействие осуществляется в зашифрованном виде без возможности корректировки цепочки серверов.

Рис. 9.2. Метод управления трафиком на основе микс-узлов

По сравнению с полностью распределенными системами данный метод имеет преимущество в более высокой скорости «интернет-серфинга» (под данным термином понимают посещение веб-сайтов, поиск и работу с информацией в сети Интернет). Вместе с тем узел клиента не выступает конечным звеном цепи, т. е. от его имени злоумышленник не сможет действовать в рамках данной сети.

9.4. Оверлейная сеть *Tor*

Знакомство с «глубоким» Интернетом стоит начать с метода управления трафиком на основе луковой маршрутизации *Tor* (англ. *The Onion Router*). Используется система прокси-серверов, позволяющая устанавливать анонимное сетевое соединение (проиллюстрировано на рис. 9.3). Клиент случайным образом выбирает три прокси-сервера (*нода*), обменивается с ними ключами шифрования и перед отправкой информации в сеть производит многоуровневое шифрование (от 3-го к 1-му ключу) каждого пакета. Промежуточные ноды могут дешифро-

вать лишь свой слой, получив из полезной информации лишь адрес следующего отправления.

Рис. 9.3. Метод управления трафиком на основе луковой маршрутизации

Таким образом, промежуточные узлы обрабатывают трассировочные инструкции и не знают адреса отправителя и получателя, а также содержание сообщения.

9.5. Проект *I2P*

Не менее содержательной разработкой является проект «невидимый Интернет» *I2P* (англ. *Invisible Internet Project*) – защищенная, анонимная, самоорганизующаяся, распределенная оверлейная сеть. Имеет свой собственный стек протоколов, работающий поверх модели *TCP/IP*. Сеть предоставляет приложениям транспортный механизм для анонимной и защищенной пересылки сообщений. Используется модифицированный *DHT Kademlia* с хранением хешированных адресов хостов сети, зашифрованных *AES IP*-адресов, *ND* и публичных ключей шифрования. Технология заслуживает досконального рассмотрения на всех уровнях функционирования. Но в целях краткости стоит привести лишь ключевую концепцию метода управления трафиком *I2P*, основанную на комбинированном туннелировании (рис. 9.4). Входящие туннели призваны отправлять датаграммы от создателя туннеля, а исходящие туннели отвечают за доставку датаграмм создателю туннеля. Цепочка односторонняя.

Если комбинировать два туннеля, узлы «А» и «В» могут обмениваться сообщениями. Отправитель «А» устанавливает исходящий тун-

нель, а получатель «B» – входящий. Шлюз входящего туннеля может получать сообщения от любого пользователя и посылать сообщения хосту «B». Оконечная точка исходящего туннеля необходима для посылки сообщения шлюзу входящего туннеля. С этой целью узел «A» добавляет инструкции к своему зашифрованному сообщению. Соответственно при дешифровке датаграммы в конечной точке исходящего туннеля извлекаются инструкции переадресации сообщения нужному шлюзу входящего туннеля хоста «B».

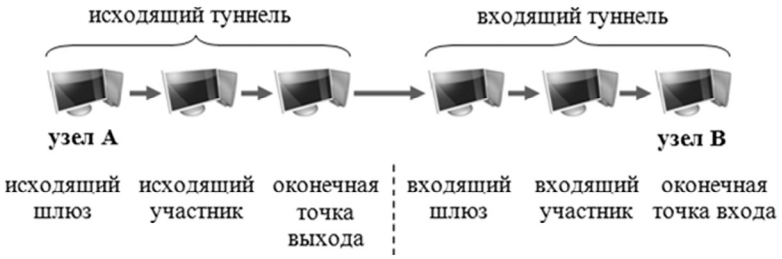


Рис. 9.4. Метод управления трафиком *I2P*, формирование туннеля

В *I2P* используется распределенная сетевая база данных в целях хранения и совместного использования сетевых метаданных, разделяемых на две категории: "*routerInfo*" и "*leaseSets*" (подписываемые одним из партнеров и верифицируемые оппонентом). Первые предоставляют информацию пограничным узлам для взаимодействия с определенным маршрутизатором (публичные ключи, транспортные адреса и др.). Вторые обеспечивают маршрутизаторы данными для взаимодействия с определенными объектами узлов назначения. Параметры *leaseSet* идентифицируют шлюз туннеля, позволяющего достичь узел назначения. Для минимизации рисков неавторизованного раскрытия имени партнера в сети добавляется еще один уровень шифрования между оконечными хостами [60–64].

9.6. Операционная система *Tails*

Tails (англ. *The Amnesic Incognito Live System*) является операционной системой, цель которой – анонимизация пользователей и обеспечение неприкосновенности частной жизни. Как и *Kali Linux*, *Tails* основана на ОС *Linux Debian*, однако *Tails* может быть установлена в качестве замены основного дистрибутива. В данной ОС предусмотрен широкий спектр механизмов обеспечения анонимности. К примеру,

после перезагрузки системы все действия пользователя автоматически стираются, что не дает возможности установить, какого рода деятельность он вел, а содержимое буфера обмена шифруется встроенными в систему средствами для его сокрытия.

Перечислим наиболее известные программные средства для обеспечения безопасности и анонимности в *Tails*:

1) *Tor* с графическим интерфейсом *Vidalia* и компонентом *Torbutton* для обеспечения защиты от вредоносных кодов *JavaScript*;

2) распределенная анонимная сеть *I2P*;

3) браузерное расширение *HTTPS Everywhere*, позволяющее получать доступ к веб-сайтам только с помощью протокола *https*;

4) модуль для защищенного мгновенного обмена сообщениями *Pidgin* в связке с расширением *OTR* (анг. *Off-The-Record*). Стоит отметить, что по умолчанию диалоги архивируются, поэтому данную опцию необходимо дезактивировать;

5) программа для шифрования данных и создания электронных цифровых подписей *GnuPG*;

6) программа шифрования *TrueCrypt*;

7) генератор надежных паролей *PWGen*;

8) инструмент анонимизации метаданных *MAT* (дата создания, координаты *GPS*, модель фотокамеры, параметры создания снимка и т. д.);

9) виртуальная клавиатура *Florence*, позволяющая предотвращать отслеживание вводимых данных с помощью кейлоггеров (англ. *keylogger*).

Однако все же стоит отметить, что все рассмотренные проекты, в том числе и *Tails*, не лишены уязвимостей, которые необходимо выявить в ходе выполнения практического задания.

9.7. Практическое задание

Необходимо изучить технологии анонимизации и защиты процесса передачи данных в глобальной сети Интернет. Затем выявить уязвимости алгоритмов и методов работы рассмотренных технологий, а также слабые места их программной реализации.

Следующим этапом необходимо исследовать инструменты «интернет-серфинга». Произвести сравнительный анализ браузеров и их надстроек. Составить исчерпывающий комплект ПО для безопасной и анонимной работы с информацией в сети Интернет. При этом не стоит забывать о проверке своих средств защиты специализированными программами и сервисами (например, "*WITCH*?" и *2ip.ru*).

Содержание отчета

1. Цель работы.
2. Техническое задание.
3. Исследование технологий анонимизации и обеспечения конфиденциальности передаваемой информации.
4. Идентификация уязвимостей.
5. Исследование и выбор инструментов «интернет-серфинга».
6. Заключение.
7. Список литературы.

Контрольные вопросы

1. Какие существуют способы анонимизации в глобальной сети Интернет?
2. Назовите основные отличия реализаций технологий виртуальных частных сетей *OpenVPN* и *IPsec*.
3. Что представляют собой механизмы проксирования? Для чего они применяются?
4. Какие сети называют оверлейными?
5. Назовите основные принципы работы сети *JAP*.
6. Назовите основные принципы работы сети *Tor*.
7. Назовите основные принципы работы сети *I2P*.
8. В чем состоят ключевые отличия рассмотренных оверлейных сетей?
9. Перечислите уязвимости рассмотренных проектов.
10. Что представляет собой *Tails* ОС? Какими особенностями обладает?

10. ПОИСК И ИЗУЧЕНИЕ УЯЗВИМОСТЕЙ НУЛЕВОГО ДНЯ

Ключевые слова: 0day, back doors, Tor, I2P.

Настоящий раздел посвящен приобретению навыков самостоятельного поиска, обработки и анализа информации в «глубоком» Интернете. Рассматриваются актуальные уязвимости программного обеспечения и инструменты скрытого управления компьютером.

10.1. Теоретические сведения

Под уязвимостью нулевого дня (англ. *zero day, 0day*) понимают термин, обозначающий неустраненные уязвимости программного обеспечения. Проблема может быть обусловлена как ошибками программирования, так и неточностью алгоритмов и методов работы объекта.

Уязвимость или атака становятся публично известны до момента выпуска производителем ПО исправлений ошибки (обновлений, патчей, заплаток). Соответственно детектировать подобные атаки традиционными сигнатурными средствами защиты невозможно. Одним из допустимых решений проблемы является использование средств защиты на основе поведенческих и интеллектуально-адаптивных методов обеспечения информационной безопасности.

Наиболее критичными уязвимостями являются инструменты скрытого управления компьютером, именуемые бэкдорами. Под угрозой оказывается не только неприкосновенность частной жизни, но и потенциальная опасность осуществления злоумышленных действий от чужого имени (атаки с имперсонацией). К сожалению, стоит отметить, что часто сами производители программного обеспечения интегрируют бэкдоры в свои продукты, руководствуясь при этом коммерческими или политическими целями.

10.2. Практическое задание

Необходимо произвести поиск и изучение уязвимостей нулевого дня, ранее не опубликованных в глобальной сети Интернет, в том числе на официальном сайте производителя и ресурсах *xakep.ru* и *habrahabr.ru*.

Требуется продемонстрировать навыки поиска, обработки и анализа информации в «глубоком» Интернете. Рекомендуется использовать скрытые технические информационные ресурсы *Tor* и *I2P*.

Альтернативный вариант выполнения задания – самостоятельный анализ и выявление уязвимостей/бэкдоров какого-либо программного продукта.

Содержание отчета

1. Цель работы.
2. Техническое задание.

3. Поиск/выявление объекта исследования.
4. Анализ найденных уязвимостей нулевого дня.
5. Исследование и разработка средств защиты.
6. Заключение.
7. Список литературы.
8. Листинг.

Контрольные вопросы

1. Что называют уязвимостью нулевого дня?
2. Чем могут быть обусловлены подобные уязвимости?
3. Какие существуют способы обнаружения атак, использующих данную уязвимость?
4. Дайте определение понятию «бэкдор».
5. Перечислите свойства бэкдоров.
6. Каким образом можно самостоятельно выявить уязвимости нулевого дня и бэкдоры в программных продуктах?
7. Почему рекомендуется использовать скрытые технические информационные ресурсы *Tor* и *I2P* для поиска ранее неопубликованных уязвимостей?

ЗАКЛЮЧЕНИЕ

В учебном пособии были рассмотрены концептуальные принципы системного администрирования и обеспечения комплексной информационной безопасности информационных систем и корпоративных вычислительных сетей. Представлен теоретический материал, который помогает приобрести практические навыки по проектированию сетей, настройке и конфигурированию программных, управляемых аппаратно-программных средств корпоративного уровня. Рассмотрена работа системного администратора и инженера информационной безопасности в наиболее распространенных операционных системах семейства Windows и Linux.

Проанализирована работа шлюзов, веб- и файловых серверов, межсетевых экранов защиты веб-приложений, систем обнаружения и предотвращения вторжений; приобретены базовые навыки по основам администрирования централизованной работы пользователей и технологиям виртуализации. Систематизированы уязвимости управления трафиком вычислительной сети, дан обзор решения вопросов анонимизации и безопасного интернет-серфинга. Частично были затронуты криптография и практическая сторона обеспечения конфиденциальности как на локальной стороне, так и при сетевом взаимодействии.

Автор пособия искренне надеется, что читатель сумел выработать умение по поиску, обработке, систематизации и анализу информации, так как в жизни необходимо научиться принимать взвешенные, конструктивные и аргументированные решения, в минимальные сроки изучать материал на концептуальном уровне и оперативно решать поставленные задачи. Тезис о невозможности достижения безупречного уровня информационной безопасности теперь должен восприниматься конструктивно. Необходимо находить компромисс с точки зрения рентабельности и целесообразности.

Данное пособие является результатом совместной плодотворной работы Научно-исследовательского института информационно-коммуникационных технологий и Новосибирского государственного технического университета.

Имея в виду международные тенденции в области политической цензуры и тотального контроля, не стоит забывать, что изобретение различных алгоритмов и методов обеспечения информационной безопасности и анонимизации, разработка инновационных ИТ решений и публикации их с открытым исходным кодом – приносят пользу обществу.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Олифер В.Г.* Компьютерные сети. Принципы, технологии, протоколы [Текст] / В.Г. Олифер, Н.А. Олифер: учеб. для вузов. – 4-е изд. – Санкт-Петербург: Питер, 2010. – 944 с.

2. *Олифер В.Г.* Сетевые операционные системы [Текст] / В.Г. Олифер, Н.А. Олифер: учеб. для вузов. – 2-е изд. – Санкт-Петербург: Питер, 2009. – 669 с.

3. *Колесниченко Д.Н.* Linux. Полное руководство [Текст] / Д.Н. Колесниченко. – СПб.: Наука и техника, 2006. – 784 с.

4. *Басыня Е.А.* Самоорганизующаяся система управления трафиком вычислительной сети [Текст] / Е.А. Басыня, Г.А. Французова, А.В. Гунько // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2014. – № 1 (31). – С. 179–184.

5. *Basinya E.A.* Methods of self-organization in providing network security [Text] / E.A. Basinya, G.A. Frantsuzova, A.V. Gunko // Global Science and Innovation: materials of the 1 intern. sci. conf., USA, Chicago, 17–18 Dec. 2013. – Chicago: Accent Graphics communications 2013. – Vol. 2. – P. 386–389.

6. Research in Attacks, Intrusions and Defenses: 17th International Symposium, RAID 2014, Gothenburg, Sweden, September 17–19, 2014. – P. 465.

7. CRC Handbook of Financial Cryptography and Security. – London: Chapman & Hall, 2010. – P. 612.

8. Detecting distributed network traffic anomaly with network-wide correlation analysis [Text] / Li Zonglin [et al.] // EURASIP Journal on Advances in Signal Processing. – 2009. – Vol. 2009. – Art. 2. – (Special issue on signal processing applications in network intrusion detection systems).

9. Invisible Internet Project: [Электронный ресурс]. – URL: https://geti2p.net. (Дата обращения: 31.08.2015).

10. The Onion Router: [Электронный ресурс]. – URL: https://www.torproject.org. (Дата обращения: 31.08.2015).

11. Журнал по информационной безопасности, разработки ПО, DevOps: [Электронный ресурс]. – URL: https://xakep.ru. (Дата обращения: 31.08.2015).

12. Информационный ресурс для IT-специалистов: [Электронный ресурс]. – URL: http://habrahabr.ru. (Дата обращения: 31.08.2015).

13. Международный форум по практической информационной безопасности: [Электронный ресурс]. – URL: http://www.phdays.ru. (Дата обращения: 31.08.2015).

14. Cisco visual networking index: forecast and methodology, 2013–2018 [Electronic resource] // Cisco VNI. – San Jose, 2014. – URL: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.html. – Title from screen.

15. Digital attack map: top daily DDoS attack worldwide [Electronic resource] / Arbor Networks. – Burlington, 2014. – URL: http://www.digitalattackmap.com. – Title from screen.

16. 2014 McAfee report on the global cost of cybercrime [Electronic resource] / CSIS-Center for Strategic and International Studies. – Washington, 2014. – URL: http://csis.org/event/2014-mcafee-report-global-cost-cybercrime. – Title from screen.

17. *Норткат С.* Обнаружение нарушений безопасности в сетях [Текст] / С. Норткат, Дж. Новак. – Москва: Издат. дом «Вильямс», 2003. – 448 с.

18. *Кучерявый Е.А.* Управление трафиком и качество обслуживания в сети Интернет [Текст] / Е.А. Кучерявый. – Санкт-Петербург: Наука и техника, 2004. – 336 с.

19. *Иваненко Р.В.* Оптимизация пропускной способности узла в сетях с пакетной коммутацией [Текст] / Р.В. Иваненко, Р.Р. Иваненко, Л.В. Воробьев // Вестник Рязанского государственного радиотехнического университета. – 2011. – № 35. – С. 123–126.

20. *Камаев В.А.* Методология обнаружения вторжений [Текст] / В.А. Камаев, В.В. Натров // Известия Волгоградского государственного технического университета. – 2006. – № 4. – С. 148–153.

21. *Биячуев Т.А.* Безопасность корпоративных сетей [Текст]: учеб. пособие / Т.А. Биячуев; под ред. Л.Г. Осовецкого. – Санкт-Петербург: СПбГУ ИТМО, 2004. – 161 с.

22. Оценка защищенности информационно-телекоммуникационных систем, подвергающихся DDOS-атакам [Текст] / Г.А. Остапенко, М.В. Бурса, Н.И. Баранников, И.Л. Батаронов // Информация и безопасность. – 2013. – Т. 16, № 4. – С. 496–497.

23. *Ручкин В.Н.* Анализ сетевого трафика нелегитимных пакетов DOS атак [Текст] / В.Н. Ручкин, Е.А. Богданова // Информатика и прикладная математика: межвуз. сб. науч. тр. – 2012. – № 18. – С. 87–91.

24. *Французова Г.А.* Самоорганизующаяся система управления трафиком вычислительной сети: метод противодействия сетевым угрозам [Текст] / Г.А. Французова, А.В. Гунько, Е.А. Басыня // Программная инженерия. – 2014. – № 3. – С. 16–20.

25. Рабочий эксплойт для сегодняшней уязвимости CVE-2014-0160 [Электронный ресурс]. – Режим доступа: http://xakep.ru/news/62329/. – Загл. с экрана.

26. В протоколах OAuth и OpenID обнаружена уязвимость [Электронный ресурс]. – Режим доступа: http://www.securitylab.ru/news/452448.php/. – Загл. с экрана.

27. *Басыня Е.А.* О перспективах развития криптографии [Текст] / Е.А. Басыня, Г.А. Французова, А.В. Гунько // Перспективное развитие науки, техники и технологий: материалы 3-й междунар. науч.-практ. конф. В 3 т. – Курск: Изд-во ЮЗГУ, 2013. – Т. 1. – С. 199–200.

28. *Mulazzani M.* Anonymity and monitoring: how to monitor the infrastructure of an anonymity system [Text] / M. Mulazzani, M. Huber, ER. Weippl // IEEE Transactions on Systems, Man and Cybernetics. C: Applications and Reviews. – 2010. – Vol. 40, iss. 5. – P. 539–546.

29. Practical attacks against the I2P network [Text] / Chr. Egger, J. Schlumberger, Chr. Kruegel, G. Vigna // Research in Attacks, Intrusions, and Defenses: proc., 16 intern. symp., RAID 2013. – 2013. – P. 432–451. – (LNCS; vol. 8145).

30. *Basinya E.A.* Design and study of the system of computer network traffic using the secure virtual communication channel technology / E.A. Basinya, A.V. Safronov // Actual problems of electronic instrument engineering. Materials of the XIII international scientific-technical conference in 12 Volumes. RF, Novosibirsk, NSTU publ. 3–6 Oct. 2016. – Vol. 1, Part 3. – P. 132–134.

31. *Басыня Е.А.* Автоматизированная установка и конфигурирование серверных решений / Е.А. Басыня, М.С. Лукина // Современные материалы, техника и технологии. – 2016. – № 2 (5). – С. 21–26.

32. *Schomburg J.* Anonymity techniques – usability tests of major anonymity networks [Text] / J. Schomburg // Extended abstracts of the Fourth Privacy Enhancing Technologies Convention (PET-CON 2009.1). – Dresden: TU, Fak. Informatik, 2009. – P. 49–58. – (Technische Berichte).

33. *Басыня Е.А.* Алгоритмы управления трафиком в оверлейной сети I2P / Е.А. Басыня, И.В. Головченко // Фундаментальные и прикладные исследования в современном мире: материалы 14-й междунар. науч.-практ. конф. – Санкт-Петербург: ИИУНЦ «Стратегия будущего», 2016. – Т. 1. – С. 97–102.

34. *Ruiz-Martínez A*. A survey on solutions and main free tools for privacy enhancing Web communications [Text] / A Ruiz-Martínez // Journal of Network and Computer Applications. – 2012. – Vol. 35, iss. 5. – P. 1473–1492.

35. *Басыня Е.А.* Безопасность и анонимизация автоматизированной настройки серверных решений / Е.А. Басыня, М.С. Лукина // Высокие технологии и инновации в науке: материалы конф. ГНИИ «Нацразвитие»: сб. избр. ст. – Санкт-Петербург: ГНИИ «Нацразвитие», 2016. – С. 69–76.

36. *Murdoch St.* J. Low-cost traffic analysis of Tor [Text] / St. J. Murdoch, G. Danezis // IEEE symposium on security and privacy (IEEE S&P 2005): proc., USA, Oakland, 8–11 May 2005. – [USA]: IEEE, 2005. – P. 183–195.

37. *Басыня Е.А.* Разработка и исследование системы управления метаданными изображений / Е.А. Басыня, А.В. Сафронов // Актуальные проблемы электронного приборостроения. Труды XIII международной научно-тех-

нической конференции в 12 т. – Новосибирск: Изд-во НГТУ, 2016. – Т. 9. – С. 155–157.

38. Theoretical analysis of the performance of anonymous communication system 3-mode net [Text] / K. Kono, S. Nakano, Y. Ito, N. Babaguchi // IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences. – 2010. – Vol. E93-A. – № 7. – P. 1338–1345.

39. *Басыня Е.А.* Разработка и исследование системы управления трафиком вычислительных сетей с использованием технологии защищенных виртуальных каналов связи / Е.А. Басыня, А.В. Сафронов // Актуальные проблемы электронного приборостроения. Труды XIII междунар. науч.-техн. конф. в 12 т. – Новосибирск: Изд-во НГТУ, 2016. – Т. 10. – С. 163–165.

40. *Wiangsripanawan R.* Design principles for low latency anonymous network systems secure against timing attacks [Text] / R. Wiangsripanawan, W. Susilo, R. Safavi-Naini // Proceedings of the fifth Australasian symposium on ACSW frontiers, ACSW '07. – Darlinghurst, 2007. – Vol. 68. – P. 183–191.

41. *Басыня Е.А.* Разработка модуля системы обнаружения и предотвращения вторжений / Е.А. Басыня, Ю.К. Равтович // Современные материалы, техника и технологии. – 2016. – № 2 (5). – С. 26–32.

42. *Басыня Е.А.* Вопросы управления трафиком в оверлейных сетях / Е.А. Басыня // Автоматика и программная инженерия = Automatics & Software Enginery. – 2014. – № 3 (9). – С. 29–32.

43. *Timpanaro J.P.* Improving content availability in the I2P anonymous file-sharing environment [Text] / J.P. Timpanaro, I. Chrisment, O. Festor // Cyberspace Safety and Security: proc. of the 4 intern. symp. on cyberspace safety and security, Australia, Melbourne, December 2012. – Melbourne: IEEE, 2012. – P. 77–92. – (LNCS; vol. 7672).

44. *Басыня Е.А.* Метод управления трафиком на межсетевых узлах локальных вычислительных сетей / Е.А. Басыня // Известия Самарского научного центра Российской академии наук. – 2014. – Т. 16, № 4 (3). – С. 507–511.

45. *Schimmer L.* Peer profiling and selection in the I2P anonymous network [Text] / Lars Schimmer // Extended abstracts of the fourth privacy enhancing technologies convention (PET-CON 2009.1). – Dresden: TU, Fak. Informatik, 2009. – P. 59–70. – (Technische Berichte).

46. *Басыня Е.А.* О шифровании и анонимизации в вопросах обеспечения информационной безопасности / Е.А. Басыня, Г.А. Французова, А.В. Гунько // Компьютерные технологии в науке, производстве, социальных и экономических процессах: материалы 14-й междунар. науч.-практ. конф., Новочеркасск, 12 дек. 2013 г. – Новочеркасск: ЮРГПУ(НПИ), 2014. – С. 165–168.

47. *Басыня Е.А.* Проблематика управления трафиком вычислительной сети с коммутацией пакетов на основе стека протоколов TCP/IP / Е.А. Басыня // Техника и технология: новые перспективы развития. – 2014. – № 15. – С. 48–57. – Работа выполнена при финансовой поддержке Минобрнауки России по государственному заданию № 2014/138, тема проекта «Новые структуры,

модели и алгоритмы для прорывных методов управления техническими системами на основе наукоемких результатов интеллектуальной деятельности».

48. *Filiol E.* Dynamic cryptographic backdoors. Pt. 2. Taking control over the TOR network: slides [Electronic resource] / E. Filiol, O. Remi-Omosowon, L. Mutembei; ESIEA – Laval, Operational Cryptology and Virology Laboratory // The 28 chaos communication congress, Berlin, 2011. – Berlin, 2011. – URL: https://events.ccc.de/congress/2011/Fahrplan/attachments/1999_slides_28C3.pdf. – Title from screen.

49. *Басыня Е.А.* Самоорганизующаяся система управления трафиком сети: удаленный сетевой доступ / Е.А. Басыня, Г.А. Французова, А.В. Гунько // Автоматика и программная инженерия. – 2014. – № 1 (7). – С. 9–12.

50. *Платонов В.В.* Программно-аппаратные средства обеспечения информационной безопасности вычислительных сетей [Текст] / В.В. Платонов. – М.: Академия, 2006. – 240 с.

51. *Басыня Е.А.* Интелектуально-адаптивные методы обеспечения информационной сетевой безопасности / Е.А. Басыня, А.В. Гунько // Автоматика и программная инженерия. – 2013. – № 1 (3). – С. 95–97.

52. *Французова Г.А.* Обеспечение информационной безопасности внутренних информационных потоков корпоративной сети / Г.А. Французова, А.В. Гунько, Е.А. Басыня; науч. рук. Г.А. Французова // Наука. Технологии. Инновации: материалы Всерос. науч. конф. молодых ученых, Новосибирск, 21–24 нояб. 2013 г.: в 10 ч. – Новосибирск: Изд-во НГТУ, 2013. – Ч. 2. – С. 41–43.

53. *Басыня Е.А.* Оптимальное регулирование пропускной способностью вычислительной сети самоорганизующейся системой управления трафиком / Е.А. Басыня, Г.А. Французова, А.В. Гунько // Современные тенденции в образовании и науке: сб. науч. тр. по материалам междунар. науч.-практ. конф., 31 окт. 2013 г.: в 26 ч. – Тамбов: Бизнес-Наука-Общество, 2013. – Ч. 5. – С. 13–14.

54. *Французова Г.А.* Применение искусственного интеллекта в сфере сетевой информационной безопасности / Г.А. Французова, А.В. Гунько, Е.А. Басыня // Искусственный интеллект: философия, методология, инновации: сб. тр. 7 Всерос. конф. студентов, аспирантов и молодых ученых, Москва, 13–15 ноября 2013 г. – М.: Радио и связь, 2013. – Ч. 2. Секции 4–6. – С. 110–115.

55. *Французова Г.А.* Разработка и исследование самоорганизующейся системы управления трафиком вычислительной сети / Г.А. Французова, А.В. Гунько, Е.А. Басыня; науч. рук. Г.А. Французова // Наука. Технологии. Инновации: материалы Всерос. науч. конф. молодых ученых, Новосибирск, 21–24 нояб. 2013 г.: в 10 ч. – Новосибирск: Изд-во НГТУ, 2013. – Ч. 2. – С. 3–7.

56. *Французова Г.А.* Самоорганизующаяся система управления трафиком вычислительной сети: механизмы защиты от сканирования и зондирования / Г.А. Французова, А.В. Гунько, Е.А. Басыня // Сборник научных трудов

Sworld. – 2013. – Т. 9, вып. 4. – С. 75–78. – Тема вып. «Перспективные инновации в науке, образовании, производстве и транспорте – 2013».

57. *Басыня Е.А.* Технология управления трафиком вычислительной сети на основе самоорганизующихся систем / Е.А. Басыня; науч. рук. Г.А. Французова; консультант А.В. Гунько // Новые информационные технологии в научных исследованиях (НИТ-2013): материалы 18 Всерос. науч.-техн. конф. студентов, молодых ученых и специалистов. – Рязань: Изд-во РГРТУ, 2013. – С. 181–183.

58. *Гунько А.В.* Стохастические методы обеспечения информационной сетевой безопасности / А.В. Гунько, Е.А. Басыня // Актуальные проблемы электронного приборостроения. Материалы XI Международной конференции в 7 томах. – Новосибирск: Изд-во НГТУ, 2012. Т. 7. – С. 47–49.

59. *Басыня Е.А.* Стохастические методы управления трафиком вычислительной сети с коммутацией пакетов / Е.А. Басыня, А.В. Гунько // Нелинейные динамические системы: моделирование и оптимизация управления. Сборник тезисов докладов Международной молодежной конференции. – Новосибирск: КАНТ, 2012. – С. 5–7.

60. *Basinya E.A.* Design and study of the system of computer network traffic using the secure virtual communication channel technology / E.A. Basinya, A.V. Safronov // Актуальные проблемы электронного приборостроения (АПЭП–2016) = Actual problems of electronic instrument engineering (APEIE–2016): тр. 13 междунар. науч.-техн. конф., Новосибирск, 3–6 окт. 2016 г.: в 12 т. – Новосибирск: Изд-во НГТУ, 2016. – Т. 1, ч. 3. – С. 132–134.

61. *Басыня Е.А.* Защита от фальсификации информационных ресурсов / Е.А. Басыня, С.В. Казарбин // Перспективное развитие науки, техники и технологий: сб. тр. 6 междунар. науч.-практ. конф., Курск, 20–21 окт. 2016 г. – Курск, 2016. – С. 16–19.

62. *Равтович Ю.К.* Разработка модуля системы обнаружения и предотвращения вторжений / Ю.К. Равтович; науч. рук. Е.А. Басыня // Наука. Технологии. Инновации: сб. науч. тр.: в 9 ч., Новосибирск, 5–9 дек. 2016 г. – Новосибирск: Изд-во НГТУ, 2016. – Ч. 1. – С. 69–71.

63. *Басыня Е.А.* Разработка и исследование системы управления метаданными изображений / Е.А. Басыня, А.В. Сафронов // Актуальные проблемы электронного приборостроения (АПЭП–2016) = Actual problems of electronic instrument engineering (APEIE–2016): тр. 13 междунар. науч.-техн. конф., Новосибирск, 3–6 окт. 2016 г.: в 12 т. – Новосибирск: Изд-во НГТУ, 2016. – Т. 9.

64. *Колесниченко Д.Н.* Linux-сервер своими руками / Д.Н. Колесниченко. – СПб.: Наука и техника, 2006. – 587 с.

Басыня Евгений Александрович

СИСТЕМНОЕ АДМИНИСТРИРОВАНИЕ И ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

Учебное пособие

Редактор *Л.Н. Ветчакова*
Выпускающий редактор *И.П. Бровановa*
Корректор *Л.Н. Киншт*
Дизайн обложки *А.В. Ладыжская*
Компьютерная верстка *Н.В. Гаврилова*

Налоговая льгота – Общероссийский классификатор продукции
Издание соответствует коду 95 3000 ОК 005-93 (ОКП)

Подписано в печать 05.02.2018. Формат 60 × 84 1/16. Бумага офсетная
Тираж 100 экз. Уч.-изд. л. 4,65. Печ. л. 5,0. Изд. 307/17. Заказ № 318
Цена договорная

Отпечатано в типографии
Новосибирского государственного технического университета
630073, г. Новосибирск, пр. К. Маркса, 20