**Введение**

В двадцать первом веке движущей силой и главным объектом всех отраслей человеческой деятельности становится *информация*, состояние каналов, сетей и *безопасность* серверов, которые будут основой экономического развития. К сожалению, сложные сетевые технологии достаточно уязвимы для целенаправленных атак. Причем такие атаки могут производиться удаленно, в том числе и из-за пределов национальных границ. Все это ставит новые проблемы перед разработчиками и строителями информационной инфраструктуры. Некоторые современные формы бизнеса полностью базируются на сетевых технологиях (электронная *торговля*, *IP*-телефония, сетевое провайдерство и т.д.) и *по* этой причине особенно уязвимы. Потребуется здесь и международное сотрудничество в сфере законодательства и установления барьеров для сетевых террористов. Не исключено, что придется со временем модифицировать с учетом требований безопасности некоторые протоколы и программы.

*Безопасность* представляет собой комплексное понятие, куда входят технические аспекты надежности оборудования, качество питающей сети, *уязвимость* программного обеспечения и т.д. Вы можете отключить *сеть* от *Интернет*, установить систему *RAID* для обеспечения дисковой защиты, снабдить систему надежным *UPS*, но в случае, например, пожара можете потерять базу данных, которую создавали несколько лет. Вам ведь все равно, нанесен ли *ущерб* хакером или несчастным случаем! Проектируя *сеть*, надо с самого начала учитывать все возможные угрозы, как объективные, так и субъективные.

Число ЭВМ, подключенных в *Интернет*, в 2006 году достигло миллиарда. Число WEB-серверов в 2005 году перевалило за 80 миллионов. Сегодня трудно представить себе фирму, организацию или учреждение, где бы для обработки документов, ведения бухгалтерии, учета, обмена сообщениями, доступа к информационным и поисковым серверам и так далее не использовали машин, подключенных к сети. Огромная масса людей не может себе представить жизнь без доступа к сети *Интернет*, который стал еще одним средством массовой информации. Но преимущества доступа к информации через *сеть* все чаще омрачаются атаками *вирусов*, червей, троянских коней, *spyware* и хакеров.

Первый случай атаки с помощью сетевого червя зарегистрирован в *ARPANET* (*Advanced* Research Projects *Agency* *Network*) в 1988 году (Morris *worm*), что изменило сетевую среду радикально и навсегда. *Сеть* *ARPANET* официально была объявлена *Internet* в 1989 году. Первая *программа*- *sniffer* была создана 1994 году. Первые атаки на систему *DNS* произошли в 1995 году, тогда же была впервые использована *фальсификация* адреса отправителя. Гонка в сфере *DoS*-атак началась в 1996 году.

В 2004 году зарегистрировано 523.109 сетевых инцидентов (это явилось результатом мониторинга 450 сетей в 35 странах мира). Из них 41% атак соответствовали различным видам несанкционированной активности, 26% были попытками несанкционированного доступа, 21% представляли собой сканирования, 9% — *DoS*-атаки и 3% — попытки некорректного использования приложений. Особенно популярно в последнее время использование уязвимостей браузеров. За последние шесть месяцев 2005 года выявлено более 1000 новых сетевых червей и *вирусов* (см., например, [http://www.acmqueque.com](http://www.acmqueque.com/)). Большинство *вирусов* и червей с самого начала делаются полиморфными и метаморфными. Разработчиками предпринимаются специальные меры *по* их маскировке. Некоторые виды этих вредоносных программ, попав в ЭВМ жертвы, выявляет возможные уязвимости (результат такого исследования может быть передан автору программы *по* *IRC*-каналам) для последующих атак с привлечением других средств. Появились даже черви, которые уничтожают другие вредоносные программы в ЭВМ-жертве.

Внутренняя структура и функциональность *вирусов* и червей существенно усложняется. Они уже могут подбирать для себя специфические объекты для атаки. Для этих целей *вирусы* и черви используют информационные ресурсы, например, поисковые системы типа Google — таким образом они уменьшают *вероятность* обнаружения и повышают число жертв, так как время от момента разработки до детектирования и выработки средства противодействия увеличивается.

Если раньше большинство атак имели своей целью мелкое хулиганство или спортивный интерес, теперь все больший *процент* атак преследует преступные цели и коммерческий интерес. Хакеры готовят и продают наборы взломанных ими ЭВМ для рассылки *SPAM*, фишинга, шантажа и для *DoS*-атак. На данном этапе эти атаки предпринимаются против полулегального бизнеса (азартные игры через *Интернет*, порнографические сайты и т.д.). В последнее время отмечается тенденция применения сетевых атак с политической мотивацией. Данная тенденция означает, что средства вторжения будут выполнены все более профессионально.

*Безопасность* — вещь относительная. Для того, чтобы конкретизировать обсуждаемый предмет, введены четыре уровня безопасности, обозначенные латинскими буквами A-D. Каждый из уровней поделен на субуровни, обозначаемые соответствующей буквой и цифрой (например, А1, А2, А3 или D1, D2 и т.д.). Уровню D1 соответствует *DOS*, где *пользователь* имеет *доступ* ко всем системным ресурсам и программам. Владельцем файлов является текущий *пользователь*. С-уровень предлагает большую *безопасность*, чем D-уровень. На С-уровне *пользователь* должен быть идентифицирован, прежде чем он получит *доступ* к своим файлам. Стандартная система *UNIX* с канонической *схемой доступа* (login/password) относится к уровню безопасности С1. Уровень С2 включает возможность блокировать для пользователя *исполнение* некоторых команд, если не выполняются определенные условия, при этом *пользователь* не может также контролировать некоторые *операции*. Многие современные *UNIX*-системы, в частности, *SCO*-*UNIX*, могут обеспечивать *безопасность* на уровне С2. В-уровень дает еще большие гарантии безопасности, запрещая пользователю изменять условия доступа к файлам. Очень немногие операционные системы и практически ни одна коммерчески доступная система не обеспечивают такого уровня безопасности. Но при выборе следует учитывать, что повышение надежности неизбежно уменьшает удобство пользования системой, так что обычно приходится идти на определенный *компромисс*.

Разнообразие угроз, которые подстерегают пользователя, работающего в сети, огромно. Часть из них является платой за использование сложных информационных технологий, уязвимых к внешним воздействиям, другая часть сопряжена с деятельностью людей. Некоторые угрозы носят объективный характер, например, нестабильность или низкое качество питающего напряжения, электромагнитные наводки или близкие грозовые разряды, другие могут быть связаны с невежеством или неаккуратностью самого пользователя.

Если несколько лет назад атаки на сетевые объекты совершали в основном (около 90%) хулиганы, которые таким образом пытались самоутвердиться, сейчас вторжения на серверы и рабочие станции предпринимаются уже с корыстной целью. Речь не идет о вторжениях в банки, все много прозаичнее: *хакер* взламывает большое число машин, выбирая наиболее уязвимые, и формирует *базис* для распределенных сетевых атак отказа обслуживания ( *DDoS* ) на сервис-провайдеров или серверы фирм (заказы поступают от конкурентов), или для рассылки *SPAM*, что стало в нашей стране достаточно прибыльным и относительно безопасным бизнесом.

Мы в ИТЭФ были вынуждены заниматься проблемами сетевой безопасности с 1995 года, когда несколько раз подверглись *DoS*-атакам. Сначала мы даже не знали, что происходит. Наш *маршрутизатор* CISCO-4000 был достаточно тихоходен и при потоках более 2000 пакетов в секунду блокировался. Не имея специальных средств и навыков, мы на диагностику проблемы вначале тратили более суток. Связано это с тем, что *DoS*-атаки чаще всего предпринимаются с использованием фальсификации адреса отправителя. Две атаки были предприняты с ЭВМ из нашей локальной сети, взломанной ранее. Позднее такие объекты мы научились локализовать за несколько минут. Для этого была написана специальная *программа*. *Алгоритм* этой программы доложен на конференции МаБИТ-03, МГУ, октябрь 2003 год.

Но описанные атаки были вторичными. Для нас оставалось не ясным, как взламывались ЭВМ-жертвы в нашей локальной сети. Чтобы прояснить эту проблему, мы поставили на входе сети ЭВМ, на которой сначала стояла *программа* t-*meter*, а позднее — *sniffer*. Варьируя критерии отбора пакетов, мы смогли выявить *IP*-адреса машин, с которых производится сканирование адресов и портов ЭВМ нашей сети. В настоящее время число атак из расчета на одну ЭВМ превышает 15-20/сутки.

Следует учитывать, что характер атак становится все более изощренным. Хакеры объединяются в клубы, издают журналы и продают хакерские CD. Сегодня крайне актуальным становится кооперирование их потенциальных жертв.

Число официально зарегистрированных в мире сетевых вторжений различного рода возрастает экспоненциально, о чем можно судить *по* [рис. 11.1](https://intuit.ru/studies/courses/1124/201/lecture/5219?page=2#image.11.1). (см. <http://www.cert.org/stats/>). Этот рост совпадает с ростом числа узлов в *Интернет*. Применение сетей расширяется со скоростью 20% в месяц. Внешний трафик мультинациональных организаций увеличивается со скоростью 30% в год (см. также <http://www.cuviello.com/_potrfolio/_semaphore>).

Число зарегистрированных попыток нелегального проникновения в *информационные системы* растет экспоненциально, достигнув темпа прироста 60% в год. Но нужно принимать во внимание, что существенная часть таких попыток остается незамеченной или незарегистрированной. Адекватны этому и темпы роста расходов крупных фирм на средства и системы информационной безопасности. Атаки бывают следующих классов:

* базирующиеся на дефектах протоколов, например, TCP, SMTP, HTTP или DNS;
* использующие дефекты операционной системы;
* поиск и использование слабых мест программ-приложений, включая базы данных, APACHE, а также скриптов, например, CGI;
* эксплуатирующие человеческие слабости (любопытство, алчность и пр., например, троянские кони, *spyware* );
* особую группу (частично перекрывающуюся с предшествующим пунктом) составляют *вирусы*, *сетевые черви*, кролики и пр..

К первому классу относятся и атаки типа SMURF, *ICMP* *flood* и TCPSYN *flood*. *ICMP* *flood* может применить эффект усиления на локальных широковещательных адресах. Здесь следует заметить, что для аналогичных целей хакеры могут использовать протоколы *TCP* или *UDP* (например, *порт* 7).

Разнообразие угроз, которые подстерегают пользователя, работающего в сети, огромно. Если число хакеров рано или поздно достигнет насыщения, то разнообразие атак будет нарастать, в частности, из-за возможности полиморфизма (как это происходит в случае сетевых *вирусов* – модификация сигнатуры без изменения сущности атаки). На [рис. 11.2](https://intuit.ru/studies/courses/1124/201/lecture/5219?page=2#image.11.2). сделана попытка составить классификацию существующих угроз (схему нельзя считать исчерпывающей).

Классификация такого рода всегда условна. Так, *сетевой червь* Code *Red* после инфицирования десятков и сотен тысяч узлов может инициировать синхронную попытку доступа к одной и той же WEB-странице, осуществляя *по* отношению к ней *DoS*-атаку.

Чтобы писать и эффективно поддерживать программы распознавания атак, надо знать и учитывать распределение атак *по* сигнатурам, *по* времени суток (перечень регистрируемых сигнатур может меняться для дневного и ночного времени), *по* сложности распознавания и методам противодействия. Вычислительные ресурсы ЭВМ-регистратора всегда будут критическим параметром. *По* этой причине вряд ли когда-либо можно будет позволить отслеживать все известные виды атак. Да и необходимость этого не очевидна (вряд ли сейчас нужно контролировать атаки на ОС типа *DOS*). В [таблице 11.1](https://intuit.ru/studies/courses/1124/201/lecture/5219?page=3#table.11.1) представлено распределение атак *по* определенным классам. Исходные данные брались из *базы данных* BlackIce Defender.

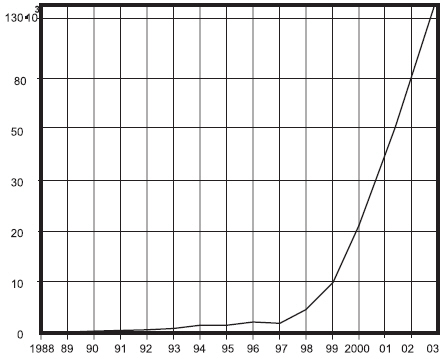
Важным фактором является актуальность *базы данных* сигнатур, кроме того, наличие сигнатуры в базе данных не означает присутствия этого типа атак во входящем трафике. Некоторые сигнатуры "выходят из моды" быстро и навсегда.

Атаки, связанные с рассылкой *вирусов*, червей, кроликов, троянских коней, требуют для достижения цели сотрудничества со стороны жертвы (используется алчность, любопытство или беспечность).

Проще всего контролировать атаки, которые реализуются одним пакетом, а характерные признаки атаки находятся в его заголовке. Но такие нападения редко остаются актуальными долго – быстро разрабатываются средства противодействия.

Практическое *значение* для конкретного сервера или сети играет реальное распределение атак *по* сигнатурам на входе системы (сервера или сети).

Сигнатуры, распознаваемые *по* заголовку пакета, составляют несколько более **15%** (здесь учтены манипуляции с заголовками *SNMP*, включая *поле* *community*, а также фальсификации *DNS*-заголовков).



**Рис. 11.1.**Распределение числа официально зарегистрированных сетевых вторжений по годам

Заметный *процент* составляют атаки, рассчитывающие достичь эффекта путем переполнения буфера (входного буфера приложения, пароля, имен и т.д.) или журнального файла, – **11.3%**. Данные разновидности атак можно отнести к *DoS*-классу.

*Некоторые сигнатуры требуют анализа не только входящего, но и исходящего трафика (например, шторм неспровоцированных откликов), что существенно усложняет алгоритм распознавания*.

Атаки на *CGI* сопряжены с генерацией нестандартных строк *URL* и параметров ( **11.5%** ), некоторые из них также работают в расчете на *переполнение* буфера параметров.

Атаки протоколов *SMTP*, *IMAP*, *POP3* и *NNTP* ( **7.8%** ) сопряжены с манипуляциями при обмене сообщениями и фальсификациями заголовков. К этому классу атак можно отнести также атаки протокола НТТР (*фальсификация* методов и заголовков объектов). Следует иметь в виду, что даже безопасные в принципе протоколы могут оказаться уязвимыми из-за некорректной реализации. Например, атакер может присылать в качестве параметра запроса "To:" нечто отличное от адреса места назначения, — при некорректной реализации почтового сервера *злоумышленник* в этом случае может получить, в частности, *доступ* к файлу *passwd* со всеми вытекающими последствиями.



**Рис. 11.2.**Схема классификации угроз

Атаки *FTP*, Telnet, *Finger*, *TFTP* обычно сопряжены с искажениями команд и их параметров. До какой-то степени эти атаки сходны с атаками на *CGI*.

В раздел "Прочие" попали атаки *IDENT*, *SQL*, сетевых печатающих устройств, *NFS*, *SOCKS*, *SMB*, *WINS*, *IIS* и т.д.

При инсталляции ОС или какого-то приложения неопытный *пользователь* может пустить процесс на самотек. При этом формируется *конфигурация* *по* умолчанию, в частности, может быть установлено *значение* пароля *по* умолчанию. *Пользователь* может полениться прочесть инструкцию, где сказано, что после завершения конфигурирования *пароль* должен быть сменен. *Хакер*, будьте уверены, инструкцию читал и с удовольствием воспользуется результатом столь легкомысленного поведения.

|  |  |  |  |
| --- | --- | --- | --- |
| Таблица 11.1. | | | |
| **Вид атаки** | | **Зарегистрированное число разновидностей** | **Доля в процентах** |
| Использование *вирусов* и червей | | 10 | 1.3 |
| Атаки уязвимостей протоколов | TCP | 15 | 2 |
| HTTP | 69 | 9.2 |
| SNMP | 79 | 10.5 |
| SMTP | 40 | 5.3 |
| DNS | 21 | 2.8 |
| SAMBA | 15 | 2 |
| Telnet | 17 | 2.2 |
| POP3 | 12 | 1.4 |
| IMAP | 6 |  |
| NNTP | 3 |  |
| Finger | 11 | 1.3 |
| FTP | 41 | 5.4 |
| TFTP | 12 | 1.4 |
| *Rlogin* | 5 |  |
| IDENT | 5 |  |
| Radius | 2 |  |
| RPC | 42 | 5.6 |
| Атаки через CGI | | 49 | 6.5 |
| Троянские кони | | >40 | 5.3 |
| DoS | | 12 | 1.6 |
| Автоматический подбор паролей (login) | | 10 | 1,3 |
| Некорректные параметры заголовков пакетов и запросов | | 22 | 2.3 |
| Сканирование портов | | 48 | 6.4 |
| Прочие | | 167 | 22 |
| Итого | | **753** | |

Атаки сканирования (*по* *IP* или номеру порта) растянуты во времени, их можно сделать трудно идентифицируемыми, замедлив темп сканирования и введя рандомизацию перебора, но они не представляют серьезной угрозы, так как всегда есть задержка между сканированием и собственно атакой (есть время подготовиться и отреагировать, например, введя *IP*-атакера в **ACL** – *Access Control* *List*).

Наиболее часто *по* данным *CERT* сканируются следующие порты (<http://www.cert.org/current/scanning.html>):

* 21/tcp ftp
* 22/tcp ssh
* 53/tcp dns
* 80/tcp http
* 111/tcp sunrpc
* 137,138,139/udp NetBIOS
* 443/tcp https
* 1433/tcp MS SQL Server

Разновидностью сканирования может являться последовательная посылка разных параметров запросов (варьироваться может как *значение* параметра, так и его *длина*). Этот вид атак часто непредсказуем, так как не известно, существует ли *значение* параметра, представляющее угрозу. Но ясно, что это *атака*, и *адрес* отправителя следует заносить в DB-атакеров и в *ACL*.

Единого определения сетевой атаки не существует, и это не удивительно. Есть сигнатуры, назначение которых не вызывает сомнения (сканирование портов "троянских коней", *ping of death*, *tear drop* и т.д.). Примерно треть разновидностей атак имеют целью выявление типа ОС, конфигурации и версий установленных приложений. Хотя непосредственно такие попытки угрозы не представляют, но это может быть разведкой перед реальной атакой. И все же более половины *сигнатур атак* столь однозначно оценить нельзя.

Рассмотрим попытку установления соединения с портом 22 (*SSH*). Когда это проделывает вычислительный *сервер* какого-то европейского научного центра, то интерпретировать попытку такого соединения как атаку вряд ли возможно, если только эта не попытка подбора пароля. Клиенты такого центра разбросаны *по* всему миру, постоянно перемещаются (входят в систему с разных *IP*-адресов), и отделить их от злокозненных хакеров практически невозможно (обычно такая *идентификация* делается на уровне аутентификации или сертификации).

Если же это *рабочая станция*, не допускающая удаленного подключения, то это, скорее всего, *атака*, особенно если такая попытка повторяется. Таким образом, одна и та же *сигнатура* в одних случаях может рассматриваться как *атака*, а в другом – как нормальное событие. Многое зависит от рабочей ОС, ее версии, набора установленных приложений (Apache, *DBI*/DBD, BIND, SendMail и пр.).

Именно это обстоятельство делает необходимым введения языков описания правил отбора и сортировки событий. С помощью этих правил *администратор*, знающий реальную конфигурацию системы, может сформулировать принципы выделения событий, сопряженных с сетевыми атаками. Это означает, что не существует и не может существовать *универсальных определений* и описаний *сигнатур атак*. Если на ЭВМ установлен *почтовый сервер*, то попытка установить с ним соединение через *порт* 25 является нормальной ситуацией, но если там такого сервера нет – это *атака* (или ошибка, если попытка не повторяется многократно). Точно так же получение *SQL*-запроса SELECT для ЭВМ-сервера DB является нормальным, а для рабочей станции – атакой. Это же можно отнести к прокси-серверам и т.д. и т.д.

Огромное число атак WEB-серверов осуществляется путем посылки нестандартных параметров запроса. Но как определить, какие параметры являются стандартными? Можно считать нестандартными параметры, содержащие непечатные символы, но этим дело не ограничивается. Здесь также мы сталкиваемся с ситуацией, когда одни и те же параметры являются нормальными для одного сервера и ненормальными (и, следовательно, указывающими на факт атаки) для другого.

*Отсюда следует, что определить, является ли данная сигнатура атакой, можно лишь, получив подробное описание конфигурации системы объекта атаки, перечня допустимых запросов и их параметров*.

Существующие *базы данных* *сигнатур атак* обобщают *опыт* регистрации атак в прошлом. *По* большей части эти атаки для новых систем угрозы уже не представляют (на них может попасться клиент, имеющий устаревшую версию ОС или приложения).

Особый *класс* сигнатур представляют *DoS*-атаки. Их многообразие растет. Следует иметь в виду, что такие атаки могут быть частью процедуры вторжения. Но их роднит друг с другом только поглощение ресурсов атакуемого объекта, *механизмы* же реализации различны. Такими ресурсами могут быть: полоса *внешнего канала*, объем памяти ЭВМ или буферов, объемы виртуальной памяти и т.д. К этой разновидности относятся *SYN*-и *ACK*-штормы, различные широковещательные запросы, *ICMP*-запросы *по* широковещательным адресам. Обычно атакер старается скрыть свой *IP*-*адрес* путем его подмены (часто это является частью схемы атаки). Атакер стремится получить усиление трафика на стороне атакуемого (конечное *значение* трафика больше потока, посылаемого атакером). Примером такого усиления могут служить запросы ресурсов *DNS*, когда пакет запроса имеет объем во много раз меньше, чем объем отклика (*IP*-*адрес* отправителя равен адресу атакуемого). Такая схема атаки эффективна, если сочетается с фальсификацией адреса отправителя (объектами атаки в этом случае становятся как *DNS*-*сервер*, так и ЭВМ, *адрес* которой атакер проставил в *поле* *адрес отправителя* ).

*Учитывая все вышесказанное, не следует стремиться создавать универсальные методы распознавания, — нужно разрабатывать легко адаптируемые методы. Программа SNORT, например, реализует именно такую модель*.

Хорошо сконфигурированная система SNORT должна регистрировать не все известные сигнатуры, а те, которые представляют потенциальную угрозу для конкретного сервера или субсети.

*По этой причине база данных сигнатур, а также набор правил для отбора случаев атак должны формироваться индивидуально для каждого сервера, для каждой субсети (в пределах субсети должны быть записаны индивидуальные правила для каждой ЭВМ)*.

Это не исключает создание программ, которые позволят унифицировать процедуру конфигурирования системы регистрации атак и противодействия им.

*Крайне привлекательным представляется создание универсальных систем распознавания атак, пригодных, в том числе, и для еще не реализованных атак (регистрация аномалий). Такие системы распознавания базируются на выявлении статистически значимого отклонения определенных параметров системы от нормы*.

Но самые эффективные атаки используют дефекты программного обеспечения атакуемой ЭВМ. Можно ли угадать, какая еще ошибка данного программного обеспечения может быть использована атакером?

Хакеры сначала испытывают различные средства локально (вне *Интернет*) и только добившись успеха (найдя реальную *уязвимость* конкретной ОС или приложения) переносят свою "работу" в *Интернет*.

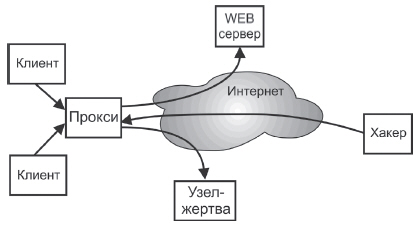
*Поэтому для обеспечения безопасности центр тяжести разработок следует перенести на сертификацию и создание средств контроля безопасности ОС и приложений*.

Все эти обстоятельства делают крайне важными подходы, использующие выявление аномалий в потоке пакетов, в поведении сети или ЭВМ.

Мало зарегистрировать атаку, надо определить и корректно интерпретировать *IP*-*адрес*, откуда эта *атака* исходит. Чаще всего, имея *IP*-*адрес*, достаточно легко отследить *путь* атаки, например, посредством *TraceRoute*. Полезным может оказаться *утилита* NSLookup, а также служба *Whois*, которая позволяет определить сервис-провайдера атакера и его географическое положение. *Анализ* потока атак показывает, что менее 40% атак осуществляется с адресов, официально зарегистрированных и имеющих имя.

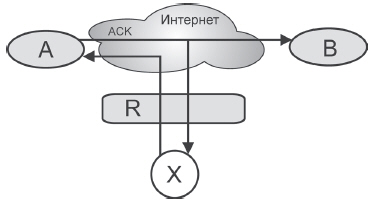
*Планируя действия против атакера, следует также учитывать, что в половине случаев (если не больше) атака предпринимается с ЭВМ-жертвы, взломанной ранее*.

Но существует *класс* атак с фальсификацией адреса отправителя, когда такую задачу решить затруднительно. Эта техника используется большинством атак отказа обслуживания (*DoS*), а также некоторыми другими (смотри RFC-2827 или <http://book.itep.ru/6/rfc2827.htm>). Два примера такого типа атак описаны на [рис. 11.3](https://intuit.ru/studies/courses/1124/201/lecture/5219?page=4#image.11.3) и [рис. 11.4](https://intuit.ru/studies/courses/1124/201/lecture/5219?page=4#image.11.4). *Атака* через прокси (например, при наличии загруженной в ЭВМ **SOCKS** программы **WinGate** ) представлена на [рис. 11.3](https://intuit.ru/studies/courses/1124/201/lecture/5219?page=4#image.11.3).



**Рис. 11.3.**

Здесь атакер ( **hacker** на рисунке) использует прокси-*сервер*, через который связаны с *Интернет* ЭВМ на левой части рисунка, для атаки машины ( **victim** ). При этом для атакуемой машины будет казаться, что ее атакует ЭВМ **SOCKS**. Особенностью этого класса атаки является то, что атакер может находиться где угодно в *Интернет*. Но в случае жалобы со стороны ЭВМ-жертвы его положение может быть локализовано.



**Рис. 11.4.**

На [рис. 11.4](https://intuit.ru/studies/courses/1124/201/lecture/5219?page=4#image.11.4) показана схема атаки с зеркалированием пакетов в маршрутизаторе R. *Маршрутизатор* **R** сконфигурирован так, чтобы копии пакетов, следующих из ЭВМ **А** в ЭВМ **В**, попадали в машину **Х** (ЭВМ атакера). В данной схеме атакер должен договориться с администратором R (если это не он сам), так как нужна специфическая *конфигурация* маршрутизатора. Но он может выйти из положения, сформировав на своей ЭВМ ложный *маршрутизатор*, поддерживающий протокол внешней маршрутизации *BGP*. Если в данной части *Интернет* *аутентификация* *BGP*-агентов соседями не производится, он легко достигнет своей цели.

Такая схема позволяет **Х** атаковать **А**, создавая впечатление, что ее атакует ЭВМ **В**. Для ЭВМ **В** будет наблюдаться эффект шторма неспровоцированных откликов со стороны ЭВМ **А** (если для атаки использован протокол ТСР).

Для распознавания сигнатуры атаки в ЭВМ **В** нужно регистрировать входящие пакеты с адресом **А** и флагом **ACK**, проверяя в то же время факт, что ЭВМ **В** не посылала **А** никаких пакетов.

Данная схема атак (с фальсификацией адреса отправителя) пригодна также и для относительно безопасной рассылки *SPAM*.

Для противодействия этому типу атак требуется тесное сотрудничество сетевых администраторов, в том числе и тех, кто не подвергается атаке. Возможна и модификация алгоритмов маршрутизаторов, которая облегчит решение проблемы (смотри ниже предложение *по* новой версии внешнего протокола маршрутизации). Сетевые администраторы должны следовать рекомендации, которая запрещает пропускать через свой внешний *шлюз* пакеты из локальной сети в *Интернет*, имеющие *адрес* отправителя, отличный от набора *IP*-адресов локальной сети. Но способа заставить выполнять такие требования в настоящее время не существует. Можно написать письма всем администраторам вдоль маршрута атаки с просьбой проанализировать трафик на предмет наличия пакетов с определенным адресом отправителя, но нельзя быть уверенным, что такая просьба будет выполнена, особенно если *администратор* так или иначе сотрудничает с атакером.

При создании протоколов *Интернет* в начале 80-х годов авторы мало думали о безопасности. Предполагалось, что пользоваться сетью будут исключительно ответственные и добропорядочные люди. Но людям, к сожалению, свойственно преувеличивать положительные качества человечества. Реальность оказалась не столь позитивна. Сказались и любовь получать что-то бесплатно, и врожденный разрушительный *потенциал*. Некоторые приемы хакеров описаны в разделе, посвященном протоколу *TCP*. Рассмотрим, что может произойти, если будет получен *TCP*-сегмент с битами *SYN* и FIN, равными 1. При получении такого сегмента протокол *TCP* осуществляет переход в состояние close\_wait. Если до этого не было установлено соединений, переход в это состояние не должен производиться, но большинство реализаций выполняют его. Такой переход крайне не желателен, так как в этом состоянии не работает *таймер* и система останется в нем вечно.

Другой вид атак протокола *TCP* называется "*syn flooding*". Здесь используется трехшаговый диалог при установлении соединения ( SYN -> syn+ack -> ack ; см. описание протокола *TCP*). Когда ЭВМ-1 получает *запрос* *syn* от ЭВМ-2, она должна подождать в частично открытом состоянии, *по* крайней мере, 75 секунд, если не придет отклик. Это позволяет успешно устанавливать *связь* даже в условиях очень больших сетевых задержек. Проблема заключается в том, что многие реализации способны отслеживать ограниченное число соединений (*по* умолчания 5). ЭВМ-*злоумышленник* может воспользоваться ограниченностью очереди и послать атакуемой ЭВМ большое число *SYN*-запросов, не отвечая на присылаемые *SYN*+*ACK*. В результате *очередь* будет быстро переполнена и прием запросов на соединение прекратится до тех пор, пока *очередь* не будет обслужена или очищена *по* тайм-ауту. Данный метод пригоден для отключения ЭВМ от сети, во всяком случае, на 75 сек. Этот вид атаки часто является частью процедуры проникновения, когда нужно заблокировать ЭВМ от получения нежелательных откликов.

Существует разновидность атаки, когда *хакер* посылает данные в пакетах с адресом отправителя, отличным от его собственного (при этом трудно установить *адрес*, откуда такая *атака* предпринимается). Здесь имеются некоторые проблемы для атакующей стороны. ЭВМ-адресат посылает все отклики *по* указанному адресу отправителя; кроме того, хакеру нужно как-то выяснить порядковый номер сегмента, записываемый в каждый пересылаемый пакет. При установлении соединения оконечный сегмент, содержащий *ACK*, должен нести *ISN* (*Initial* *Sequence Number*; cм. описание транспортного протокола *TCP*) удаленной ЭВМ. Этот сегмент посылается ЭВМ, чей *адрес* указан в первоначальном запросе *SYN*. *Хакер* должен выяснить *ISN* каким-то иным способом (этому может помочь служба netstat). Так как *ISN* — 32-разрядное число, случайно угадать его совсем не просто. Но в большинстве систем (например, *unix* bsd) *ISN* берется из некоторого счетчика, содержимое которого увеличивается на 128 каждую секунду и на 64 после каждого нового соединения. Например:

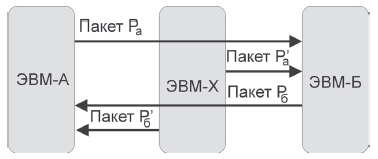
x -> s: syn(ISNx)

s -> x: syn(ISNs), ack(ISNx) (получено искомое текущее значение кода ISNs)

Таким образом, установив однажды соединение с нужной ЭВМ ( s ), некоторое время спустя можно с высокой вероятностью угадать *значение* *ISN*, послав несколько пакетов с разными значениями *ISN*. При завершении процедуры посылается сегмент *SYN*+*ACK*, который будет отвергнут, так как машина-получатель не инициализировала *связь*. В пакете отклике будет установлен флаг *RST* и соединение окажется абортированным. Чтобы этого не произошло, *хакер* может предпринять атаку типа *syn*-*flooding*, которая приведет к игнорированию *SYN*+*ACK*. В результате соединение будет установлено и *хакер*, празднуя победу, может продолжить свое черное дело, например, он может исполнить какую-либо r-команду на атакуемой ЭВМ. Если даже изменять *ISN* каждые 4 микросекунды, проблему угадывания решить не удастся. Радикальным решением может стать лишь псевдослучайная генерация *ISN*-кодов (при этом случайным образом должны задаваться не менее 16 *бит*). Решить проблему поможет *шифрование* соответствующей части содержимого пакета. Рассмотрим здесь еще несколько таких атак.

Один из известных методов базируется на *IP*-опции *"маршрутизации отправителя"* (*source routing*). ЭВМ-инициатор обмена может специфицировать *маршрут*, которым получатель должен воспользоваться при пересылке отклика. *Маршрут* может быть специфицирован так, чтобы он проходил через узел, контролируемый хакером. Эта достаточно простая методика атаки в настоящее время неэффективна, так как большинство маршрутизаторов сконфигурированы так, чтобы отфильтровывать пакеты с маршрутизацией отправителя.

Интересные возможности предоставляет вариант, когда *хакер* вставляет свою машину на пути обмена двух других узлов (hijacking). Обычные методы атак (типа *spoofing*) могут не привести к успеху из-за необходимости идентификации (*пароль*!). В данном методе *хакер* позволяет завершиться установлению связи и аутентификации и только после этого перехватывает *контроль* над виртуальным каналом. Этот способ использует механизм "десинхронизации" *TCP*-соединения. Когда порядковый номер полученного пакета не совпадает с ожидаемым значением, соединение называется десинхронизованным. Полученный пакет, в зависимости от его номера, будет выброшен или буферизован (если он находится в пределах окна). Таким образом, если узлы, участвующие в обмене сильно десинхронизованы, приходящие к ним пакеты будут отбрасываться. *Хакер* в этом случае может вводить пакеты с корректными порядковыми номерами. Разумеется, это совсем просто, если машина хакера является транзитной, что позволяет ей "портить" или удалять нормальные пакеты и подменять их своими. Но *атака* возможна и в случае, когда машина хакера находится где угодно. В этом случае "отброшенные" пакеты могут вызвать посылку *ACK*, которые содержат ожидаемое *значение* порядкового номера пакета, но и они будут отвергнуты из-за неверного их номера и т.д.. При реализации этой схемы пересылается большое число "лишних" пакетов *ACK*. Количество их будет сколь угодно велико — поэтому данная ситуация и называется "штормом *ACK*". Обмен уведомлениями (*ACK*) о неверных *ISN* будет продолжаться до тех пор, пока один из таких пакетов не будет потерян. Механизм обмена *IP*-дейтограммами устроен так, что *вероятность* потери тем выше, чем больше пакетов передается, т.е. процесс саморегулируется. Тем не менее, высокая интенсивность потока сегментов *ACK* может говорить об атаке. В нормальной ситуации *процент* сегментов *ACK* от общего потока *TCP* составляет 30-45%. Иллюстрацией данного метода нападения может служить [рис. 11.5](https://intuit.ru/studies/courses/1124/201/lecture/5219?page=5#image.11.5).



**Рис. 11.5.**

Десинхронизация может быть осуществлена при установлении соединения. *Хакер* в этом случае обрывает процесс трехшагового диалога. После того как ЭВМ **Б** пошлет *SYN*+*ACK* ЭВМ **А**, ЭВМ хакера пошлет пакет от **Б** к **А**, который с помощью бита *RST* оповещает о закрытии соединения. Далее затевается новый трехшаговый диалог с целью установления соединения с **А**, но уже с другими кодами порядковых номеров. Схема алгоритма атаки показана на [рис. 11.5](https://intuit.ru/studies/courses/1124/201/lecture/5219?page=5#image.11.5).

ЭВМ **Б** проигнорирует сообщения от **А**, так как они имеют неправильные (навязанные хакером) номера, а ЭВМ **А** проигнорирует сообщения от **Б**, так как ожидает, что они будут пронумерованы *по*-новому. Теперь *хакер* знает, какие номера нужны **А** и **Б** (непосредственно же **А** и **Б** взаимодействовать не могут, так как не знают правильных номеров), и может перехватывать, модифицировать или посылать сообщения *по* своему усмотрению как для **А**, так и для **Б**.

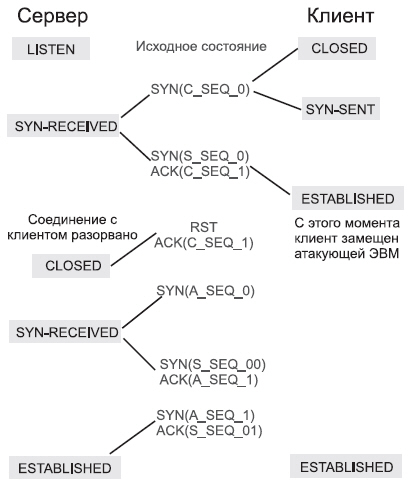
Но десинхронизация может быть реализована и в ходе сессии. Если послать флаг *RST* в середине сессии, соединение будет закрыто, о чем будет оповещено *приложение* и *пользователь*. Аналогичного эффекта можно добиться, послав отправляющей стороне ICMP\_NET\_UNREACH, ICMP\_HOST\_UNREACH или ICMP\_PORT\_UNREACH. Вариант с привлечением протокола *ICMP* привлекателен тем, что не требует угадывания *ISN*. Для того, чтобы вызвать десинхронизацию в середине сессии, не закрывая соединения, достаточно поменять порядковые номера в сообщениях. Протокол telnet имеет механизм, который позволяет решить такую задачу. В рамках протокола можно посылать команды nop ("Ничего не делать", согласитесь — хорошие команды!). Эти команды не производят никакого эффекта, но увеличивают ожидаемое *значение* порядкового номера сегмента. Послав некоторое количество таких команд, ЭВМ хакера вызовет десинхронизацию. Теперь только *хакер* знает правильные значения порядковых номеров пакетов и может начать свою подрывную работу.

Одним из наиболее эффективных методов обнаружения таких атак помимо процента сегментов *ACK* является сравнение потоков *ACK* для сервера и клиента. Полной гарантией безопасности в такой ситуации может стать только *шифрование* на прикладном уровне, например *по* схеме **kerberos**. В случае использования почты хороший результат может дать *электронная подпись* (например, в *PGP*).

*Атака* на раннем этапе установления соединения может осуществляться следующим образом. (Соединение при этом разрушается и вместо него формируется новое с другим значением *ISN*.) *Хакер* прослушивает *виртуальный канал* и ждет сообщения *SYN*+*ACK* от сервера клиенту (вторая фаза установления соединения). При получении такого сообщения *хакер* посылает серверу *RST*-*кадр*, а затем *SYN*-пакет с теми же параметрами (*TCP*-*порт*), но с другим значением *ISN* (далее и на [рис. 11.6](https://intuit.ru/studies/courses/1124/201/lecture/5219?page=5#image.11.6) этот *кадр* обозначается **a\_ack\_0** ; *префикс* А указывает на принадлежность атакующей машине хакера). *Сервер*, получив *RST*, закроет прежнее соединение и, получив *SYN*, сформирует новое для того же порта, но с новым значением *ISN*, после чего пошлет клиенту *кадр* *SYN*+*ACK*. Детектировав этот пакет, *хакер* посылает серверу *кадр* *ACK*. При этом *сервер* переходит в состояние established. Клиент перешел в это состояние при получении от сервера первого кадра *SYN*+*ACK*. На [рис. 11.6](https://intuit.ru/studies/courses/1124/201/lecture/5219?page=5#image.11.6) не показан обмен пакетами *ACK*, вызванными получением некорректных значений *ISN*. Как *сервер*, так и клиент находятся в несинхронизованном состоянии established. Обмен здесь полностью контролируется атакующей ЭВМ

Определенные услуги хакеру может оказать внутренний *протокол маршрутизации* **rip**. Этот протокол служит для рассылки информации о маршрутах в локальной сети и не требует контроля доступа, что всегда благоприятствует злоумышленникам. *Хакер* может разослать маршрутную информацию, которая убедит окружающие узлы в том, что наикратчайший *путь* во внешний мир проходит через его ЭВМ **Х**. Тогда все пакеты, адресованные за пределы локальной сети, будут сначала попадать в ЭВМ **Х**.

Хакеры часто пытаются использовать *IP*-опцию маршрутизации отправителя. Эта *опция* позволяет хакеру послать пакет извне локальной сети с адресом отправителя, соответствующим одному из внутренних адресов. Если *маршрутизатор* не отбрасывает такие пакеты, у хакера появляется шанс для атаки, при которой он находится в безопасности и дискредитирует некоторую машину локальной сети. Но *блокировка* опции маршрута отправителя еще не гарантирует отсутствия подобных атак. *Хакер* может фальсифицировать *адрес* отправителя и без этой опции, пытаясь создать иллюзию отправки пакета из *LAN*.



**Рис. 11.6.**Алгоритм атаки с использованием десинхронизации. Префикс А указывает на то, что сегмент послан атакующей ЭВМ; С — клиентом; S — сервером

Протокол управляющих сообщений **ICMP** носит вспомогательный характер (вспомним популярную утилиту *ping*, использующую этот протокол). Пакеты *ICMP* также не требуют проверки доступа (именно благодаря этому протокол пригоден для работы в масштабах *Интернет*). *По* одной этой причине протокол привлекателен для хакеров. Известны случаи, когда с его помощью люди с разных континентов бесплатно обменивались сообщениями. Пригоден протокол и для блокировки обслуживания: для этого применяются *icmp*-сообщения *"time exceeded"* (превышено время) или *"destination unreachable"* (адресат недоступен). Первое сообщение означает, что *предел*, указанный в *поле* *TTL* заголовка пакета, превышен. Такое может произойти из-за зацикливания пакетов или когда адресат расположен очень далеко. Сообщение *"destination unreachable"* имеет несколько значений. Но все они означают, что нет способа благополучно доставить пакет адресату. Оба сообщения вынуждают адресата немедленно прервать соединение. Для реализации своих коварных планов хакеру достаточно послать одно из этих сообщений одному или обоим участникам обмена.

Протокол *ICMP* может применяться и для перехвата пакетов. *ICMP*-сообщение *"redirect"* обычно используется внешними портами (*gateway*) сети в случаях, когда какая-то ЭВМ *по* ошибке полагает, что адресат находится вне локальной сети, и направляет пакеты, адресованные ему, через внешний *порт*. *Хакер* может послать *ICMP*-сообщение *"redirect"* и вынудить другую ЭВМ посылать пакеты через узел, указанный хакером. У данного метода (так же, как и предыдущего с использованием *RIP*) имеется определенное ограничение — *хакер* и жертва должны находиться в пределах общей локальной сети.

Еще одним объектом атак может стать **DNS** (*Domain Name* *Service*) локальной сети. Например, *администратор* узла *durak.dura.com* может принять решение разрешить только местные соединения. Это может быть специфицировано с помощью записи "allow \*.dura.com", а не обязательно через указание *IP*-адресов, тем более, что в сети может использоваться большое число блоков *IP*-адресов. Когда соединение установлено, ЭВМ durak обращается к *DNS*, чтобы преобразовать *IP*-*адрес* отправителя в имя, которое затем служит для проверки условия, заданного администратором. Если *хакер* имеет *доступ* к местному *DNS*-серверу, он может заставить *DNS* откликаться на его *IP*-*адрес* любым именем ЭВМ или домена. Зная об установленных администратором ограничениях, *хакер* может сделать так, что на *запрос* с его *IP*-адресом *DNS* будет откликаться именем "trustme.dura.com" (это удовлетворяет установленному ограничению). Такая *атака* может быть легко предотвращена путем повторного *DNS*-запроса с именем ЭВМ, присланным при первом запросе. Если *IP*-*адрес*, полученный при втором запросе, не совпадет с использованным в качестве параметра в первом, это означает, что *DNS* подверглась атаке.

Для атаки *DNS* извне достаточно узнать номер используемого *UDP*-порта и *ISN*. О способах определения *ISN* говорилось выше. Задача упрощается в тех случаях, когда начальным значением *ISN* является нуль. Номер же используемого в данный момент *UDP*-порта при определенном везении можно найти путем подбора, ведь здесь нет никакой аутентификации или ограничений на число попыток. *DNS* — один из самых привлекательных объектов для атак, так как через посредство AXFR-запросов можно получить информацию об узлах, которые используют определенные версии ОС с известными слабостями, упрощающими *вторжение*.

Многие атакеры применяют социальную инженерию и психологию, чтобы спровоцировать потенциальную жертву к действиям, которые нанесут *ущерб*. Например, год назад в ИТЭФ пришли письма, которые, если верить заголовку, были посланы сетевым администратором. *Приложение* к письму было заархивировано, но для дезархивации требовался *ключ*, который содержался в тексте письма. В самом письме говорилось, что *приложение*, содержит инструкцию *по* улучшению безопасности. *Почтовый сервер* ИТЭФ имеет антивирусную защиту, но зашифрованность приложения препятствовала распознаванию сигнатуры *вируса*. Хотя *здравый смысл* подсказывает, что сетевому администратору не нужно шифровать сообщение, адресованное клиентам сети, среди пользователей нашлось около десятка простаков, которые попались на эту удочку.

Потенциальную угрозу безопасности могут представлять специальные вставки во встроенное *программное обеспечение* процессора, *BIOS*, операционной системы и приложений. Аналогичные угрозы представляют любые программы с неизвестными исходными кодами, включая обеспечение маршрутизаторов. Считывать данные можно путем регистрации электромагнитного излучения дисплея и других устройств ЭВМ и даже анализируя звук при нажатии терминальных клавиш. Этим *список* способов несанкционированного доступа не исчерпывается, но прочие угрозы здесь не будут рассматриваться.

*По этой причине предпочтение следует оказывать программам с открытым кодом или коммерческим программам, где продавец документально гарантирует безопасность*.

За период с 2001 *по* 2003 годы число зарегистрированных компьютерных преступлений в России удваивалось каждый год. Считается, что 80% компьютерных преступлений не попадает в официальную статистику.

Если имеется несколько ЭВМ, объединенных в систему, и к этой системе имеет *доступ* определенное число удаленных клиентов, то наиболее уязвимыми являются соединения клиентов с этой системой. Хакеры стараются атаковать самое *слабое звено* в системе. Особенно уязвимы пользовательские, домашние ЭВМ. Зная это, некоторые банки предлагают своим клиентам удешевленные или даже бесплатные средства защиты (например, антивирусные программы). Некоторые сервис-провайдеры предоставляют своим клиентам антивирусные программы и *firewall*.

Число *сигнатур атак* и их многообразие продолжает лавинообразно увеличиваться. Происходит динамический прогресс системы щит-меч. Многие администраторы и хозяева сетей полагают, что до сих пор все обходилось, Бог даст, обойдется и впредь. Такая позиция рано или поздно приведет к серьезным потерям. Это касается не только компаний, имеющим конкурентов (о них позаботятся непременно), или структур, имеющих на серверах конфиденциальную информацию (например, государственные учреждения), но и сетей, которые, на первый взгляд, не должны быть привлекательными для хакеров. Мы за счет большого числа машин в сети убеждались в этом уже не раз. Однажды были стерты несколько сот файлов общедоступного сервера [http://book.itep.ru](http://book.itep.ru/). Казалось бы, какой в этом смысл? К счастью, у меня была свежая резервная копия...

*Прерывать работы по совершенствованию систем сетевой защиты нельзя. Рассчитывать на получение бесплатных разработок из Интернет наивно. Каждая вторая такая программа сама содержит в себе spyware* *! Поставщик антивирусной программы (особенно бесплатной версии) может быть сам разработчиком вируса* *или spyware*.

Все это делает крайне актуальными средства мониторинга и предотвращения атак, а также программы и методики анализа успешных атак и ликвидации их последствий. Рассмотрим некоторые из угроз (*по* материалам журнала *Network Security* за 2003-2005 годы и другим источникам). (Смотри также <http://book.itep.ru/6/intrusion.htm>.)

При расследовании инцидентов полезно знать базовый *инструментарий* хакеров.

*Программа* сканирования сетевых объектов (*по* *IP*, портам и пр.).

* Программа подбора паролей с каталогом заготовок (сотни мегабайт)
* Программа-sniffer ***sniffer*** (перехват пакетов в сетевом сегменте)
* Программы троянских коней и соответствующие библиотеки
* Средства выборочной модификации системных журнальных файлов
* Средства сокрытия текущей активности
* Средства автоматической модификации конфигурационных файлов.