

Александър Милев
Борислав Найденов

Администриране на мрежи

Учебното пособие е предназначено за студенти редовно и задочно обучение по специалности „Компютърни и информационни технологии”, „Компютърна информатика”, „Комуникационни и информационни системи” и магистърска специалност „Системно администриране” в Шуменски Университет „Епископ Константин Преславски” гр. Шумен, съобразно с конкретния учебен план, и сродни специалности от други висши училища в страната.

Настоящото пособие е полезно и за студентите от редица сродни специалности.

Част 5 е подготвена от Б. Найденов, а части 1, 2, 3, 4, 6, 7, 8 и 9 - от Ал. Милев.

Авторите ще приемат с благодарност всички забележки, отнасящи се до подобряването на учебния материал.

2010 г.

СЪДЪРЖАНИЕ

Увод	5
Първа глава	
Протоколен стек TCP/IP	7
1.1 Модел на TCP/IP	7
1.2 Мрежово ниво	11
1.3 Транспортни протоколи	18
1.3.1 Транспортен протокол TCP	19
1.3.2 Транспортен протокол UDP	43
Втора глава	
Заплахи за сигурността на мрежата	48
2.1 Мрежова сигурност	48
2.2 Видове заплахи за сигурността	49
2.3 Видове атаки	62
Трета глава	
Принципи на контрол на трафика	79
3.1 Структура на гранична мрежа	79
3.2 Филтриране на пакети	87
3.3 Насоки за настройване на защитни стени	93
Четвърта глава	
Мрежово администриране	98
4.1 Принципи на мрежовото администриране	98
4.2 Мрежови операционни системи	117
4.3 Директорийни услуги	124
4.4 Политики за сигурност	136
4.5 Дискови квоти	148
4.6 Защита на данните и избягване на сринове	150
Пета глава	
Технологии за сигурност	158
5.1 Основни криптографски концепции	158
5.2 Протоколи за сигурност на приложния слой	177
5.3 Протоколи за сигурност в транспортния слой	181
5.4 Защита на мрежовия слой	186
5.5 Технологии за сигурност в каналния слой	200
5.6. Модели за инфраструктура и разпространение на публични ключове	206
Шеста глава	
Виртуални частни мрежи	213
6.1 Обща положения при виртуалните частни прежи	213

6.2 VPN Сигурност	216
6.3 Примерна VPN	226
Седма глава	
Качество на услугите	232
7.1 Осигуряване качество на услугите	232
7.2 Приоритизиране на трафика	236
7.3 Практическа реализация	253
Осма глава	
Логови файлове. Системи за засичане на нарушения	274
8.1 Наблюдение и управление на логови файлове	274
8.2 Системи за засичане на нарушения	288
Девета глава	
Управление и наблюдение в мрежите	300
9.1 Управление на мрежата SNMP	300
9.2 Наблюдение, управление и отстраняване на неизправности в мрежата	312
Литература	327

Увод

Администрирането на една мрежа се явява сложен процес, който в никакъв случай не може да се ограничи в рамките на един или два дена. Персоналът отговорен за тази дейност трябва да работи винаги в екип. Но основният отговорник на мрежата е именно системния администратор.

Неговите задължения включват редица задачи, като те могат да бъдат групирани в следните направления:

- Планиране на мрежата
- Оценка заплахите за мрежата
- Създаване и поддръжка на потребителите в системите
- Управление на потребителите в мрежата и ползваните от тях ресурси
- Архивиране на данните – ежедневно, седмично, месечно
- Проверка на системните ресурси, както и на файлове, свързани с базите данни, използвани в мрежата.
- Проверка на логовите файлове на сървърите и други активни устройства като комутатори (switches), рутери (routers) и защитни стени (firewall) и др.
- Стартиране на пълно сканиране за вируси и друг зловреден код
- Проверка за обновяване на приложния софтуер
- Осъвременяване на антивирусните и други защитни системи.
- Изпълнение на процедури за цялост на системата : Scan Disk, Defragment
- Проверка и синхронизация на времето.
- Създаване на процедура за възтановяване на системите при непредвидени сривове.
- Периодична проверка производителността на мрежата
- Събиране на информация за поведението на устройствата в мрежата
- Прилагане на софтуер за регистриране на пробив в сигурността на мрежата
- Проверка на системата за пропуски в сигурността: Penetrating tests
- Обновявания и инсталиране на добавки за операционните системи

Не и на последно място е необходимостта от документиране на промените в системите и мрежата.

Очевидно е, че проблемите в администрирането на една мрежа не са малко.

Настоящото учебно пособие има за цел да се представят основните моменти от посочените дейности.

За да могат да бъдат правилно разбрани всеки един от тези елементи в администрирането на една мрежа е необходимо читателя да има основни познания по компютърни мрежи и желание да „стъпи” в необятната шир на всеки един от проблемите, които могат да възникнат в тях.

Като се имат предвид посочените особености в администрирането на една мрежа, учебното пособие разглежда в подробности протоколният стек ТСР/ІР.

На тази основа са дефинирани заплахите, които могат да се окажат фатални за всяка една мрежа.

Определени са видовете атаки за мрежите, както и мерки за борба с тях.

Разгледани са общите постановки и принципи за мрежовото администриране, свързани с потребителите и ресурсите, които са достъпни за тях.

Управлението ресурсите в мрежата и начина за тяхното ползване е също един основен въпрос, на който е обърнато внимание.

Визирайки сигурността в мрежата като основен проблем, учебното пособие се спира на основните аспекти за сигурността на различните нива, където тя може да се реализира.

Като елемент за разрешаване на проблема със сигурността, се явява комуникацията през несигурната мрежа Internet. Решението в случая е виртуалната частна мрежа (VPN), която е решение, често срещано в практиката. Принципът за действие и работа на VPN е също обект на настоящото пособие.

Качеството на услугите в една мрежа и начините за изпълнение на изискваните от потребителите качествени услуги са елемент, на който пособието се спира също в подробности.

И не на последно място се разглеждат способите за наблюдение на събитията в мрежата, тяхното автоматично регистриране и принципа за визуализиране на поведението на самата мрежа. На тази основа се дефинират и основните моменти за откриването и отстраняването на проблеми в мрежите.

Глава Първа

Протоколен стек TCP/IP

Съкращението TCP/IP (*Transmission Control Protocol/Internet Protocol*) е общо наименование на съвкупност от протоколи.

Основната цел на тази съвкупност от протоколи е да се зададе възможност за осъществяване на връзки в компютърните мрежи, предлагащи универсални мрежови услуги. В зависимост от технологията на изпълнение всяка комуникационна мрежа притежава собствен мрежов интерфейс, представен под формата на програмен интерфейс и поддържащ базови комуникационни функции, наречени примитиви. Комуникационните услуги се изграждат на основата на софтуер, който осъществява връзката между физическата мрежа и съответното приложение. По този начин сервизните функции (услугите) поддържат дефиниран интерфейс за мрежовите приложения, който е независим от намиращата се на по-ниско ниво физическа структура на мрежата. По същество архитектурата на физическото ниво е скрита за потребителя.

Друга задача, която се поставя за комуникацията между отделните системи е връзката между системи с различна физическа структура да се представи на потребителя като съвкупност от една единна мрежа. За да се даде възможност да се свържат две мрежи, е необходима компютърна система, притежаваща интерфейс и в двете. Такава система се нарича маршрутизатор (рутер, *router*). Терминът IP маршрутизатор (IP *router*) указва, че устройството, осъществяващо връзката между двете мрежи, е част от IP слоя на протоколния стек TCP/IP.

Маршрутизаторът се характеризира със следните особености:

- от гледна точка на мрежата, той е обикновена компютърна станция – хост;
- от гледна точка на потребителя, рутера е невидим. Потребителят забелязва единствено работата между крайните точки в мрежата.

Компютърна система, която е включена към мрежата се обозначава като хост. За да се идентифицира еднозначно в мрежата, на всеки хост се присъединява адрес, наречен IP адрес. Ако даден хост притежава множество мрежови интерфейсни адаптери, т.е. той участва в няколко мрежи, всеки един от тях притежава свой собствен уникален адрес. Всеки IP адрес се състои от две части – мрежов номер и адрес на отделния хост в рамката на мрежата. Мрежовият номер е част от IP адреса и се разпределя централно. Отговорността за разпределяне на отделните адреси на компютрите е в самата организация, която притежава съответната IP мрежа.

1.1 Модел на TCP/IP

Моделът TCP/IP се изгражда в съответствие с еталонния модел на слоеве. На фиг.1.1 е показано съответствието на еталонния модел с модела на TCP/IP.

Всеки един протоколен слой представлява съвкупност от мрежови функции:

- **Потребителски слой** – предоставя интерфейс за потребителските приложения, използващи TCP/IP за комуникация. Приложението представлява потребителски процес, който работи съвместно с друг процес на същия или на различен хост. Пример за такива приложения са TELNET, FTP, SMTP и др. Интерфейсът между отделните приложения и транспортния слой е дефиниран чрез комуникационни портове и сокели;

- **Транспортен слой** – реализира връзки от тип крайна точка – крайна точка, като множество приложения се обслужват едновременно. Транспортният слой отговаря за осъществяване на сигурен обмен на информация. Наличните два протокола за осигуряване на преноса са TCP и UDP (*User datagram protocol*). Протоколът UDP поддържа отделните мрежови услуги чрез несвързани с информационния поток комуникации. По този начин, приложенията използващи UDP като транспортен протокол, могат да реализират собствен контрол на потока от информация. Обикновено UDP се използва при приложения, които се нуждаят от бърз транспортен механизъм.

- **Мрежов слой** – този слой поддържа виртуално изображение на мрежата. Той представлява най-горният слой на физическата мрежова архитектура. IP (Internet protocol) е основният и най-важен протокол на този слой. Той не е свързан с отделния информационен поток. Протоколът не поддържа механизми за осигуряване на надеждни комуникации, контрол на потока данни и възстановяване при грешки. Тези функции е необходимо да се реализират на по-високо ниво на комуникация. Част от информацията, обменяна между отделните компютри, е информация за маршрутизиране. Тази информация дава възможност отделните съобщения да бъдат правилно разпределени към тяхното предназначение. Тези функции за маршрутизиране се осигуряват от IP протокола. Други по-важни протоколи на мрежово ниво са ICMP, ARP и RARP.

- **Слой на мрежовия интерфейс** – той съответства на каналния слой в еталонния модел. Дава възможност за връзка към съответната мрежова апаратна част. Мрежовият интерфейс, в зависимост от изпълнението си, може да поддържа, или не, механизми за надеждно разпределение на информационния поток. Реализацията му може да бъде пакетно-ориентирана или поточно-ориентирана. На практика TCP/IP не дефинира определен протокол за това ниво, което дава възможност за по-голяма гъвкавост на изгражданата система. Примери за изпълнение на този слой са IEEE 802.2, ATM, FDDI и др.

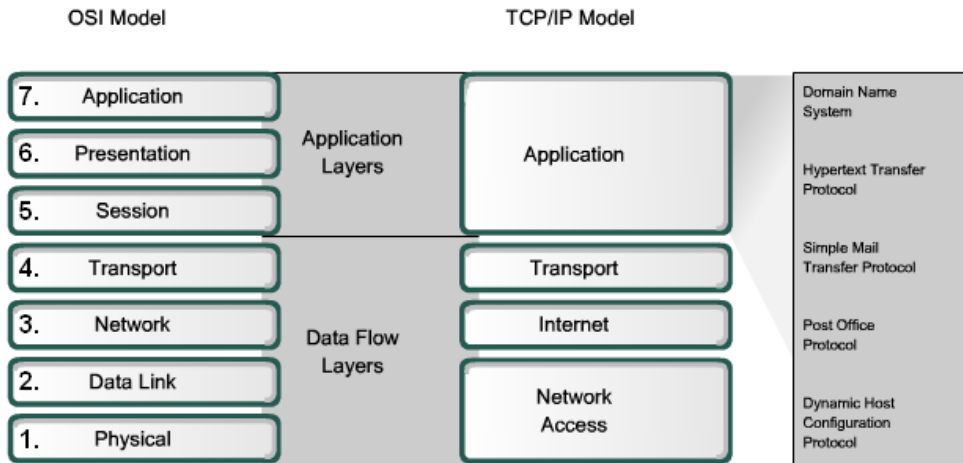
Най-високото ниво при модела TCP/IP са приложените протоколи. Те комуникират с приложения на друг мрежов хост и представляват видимите за потребителите интерфейси от съвкупността на TCP/IP протоколите.

Потребителските протоколи притежават някои общи характеристики: могат да бъдат създадени от отделни потребители, или могат да представляват стандартизирани приложения.

TCP/IP приложно ниво притежава приложни програми като:

- TELNET – осигуряващ интерактивен достъп до отдалечения хост;
- FTP (File Transfer Protocol) – осигуряващ високоскоростен трансфер на файлове от дисково устройство към друго подобно;

- SMTP (Simple Mail Transfer Protocol) – използва се като мрежова пощенска система.



Фиг. 1.1 Съответствие на модели и протоколни стекове

Това са само някои от разпространените протоколи на потребителско ниво, но съществуват и много други. Всяко отделно изпълнение на TCP/IP включва в себе си по-голяма или по-малка съвкупност от потребителски протоколи. Те използват или TCP или UDP като транспортен механизъм. Важна особеност е, че UDP е ненадежден протокол и не поддържа механизми за контрол на потока данни. В такива случаи приложението трябва да реализира собствени механизми за възстановяване при грешки и за реализация на контрол на информационния поток. В много случаи е по-лесно да се създават приложения, използващи вградения в TCP механизъм за контрол на информационния поток. Повечето приложения използват този подход, но трябва да се спомне, че използването на UDP като транспортен протокол дава възможност за по-висока производителност при осъществяване на комуникацията в следствие на редуцираната управляваща информация.

Както беше споменато TCP/IP представлява свързан с информационния поток протокол от типа крайна точка – крайна точка. Мрежовите приложения, изградени на базата на протоколния стек TCP/IP използват много често модела клиент/сървър (фиг.1.2) за комуникационен модел. Сървърът трябва да се разглежда като приложение, което предлага определени мрежови услуги (сервизни функции) на потребителите на мрежата. Клиентът е тази част от потребителската програма, която е реализирала определени заявки за обслужване от страна на сървърното приложение. Възможно е едно приложение да притежава и двете части. Тези две части могат да бъдат изпълнявани на една и съща компютърна система (хост) или на две различни системи.



Фиг. 1.2 Схема на мрежа тип клиент - сървър

Потребителите използват обикновено клиентската част на приложението, която прави заявки за отделни услуги или изпраща към сървър информация, използвайки протокола ТСП/Р като транспортно средство.

Сървърното приложение се явява програма, която получава заявка, изпълнява заявените услуги (сервизни функции) и изпраща обратно резултатите като отговор. Сървърът може да обслужва множество заявки по едно и също време.

Някои от сървърите очакват заявки на общо познатите портове, като по този начин клиентът знае предварително сокета (комбинацията IP адрес и порт за връзка), към който трябва да се отпрати заявката. От страна на клиента се използват произволно избрани портове за комуникация, докато за заявената услуга се включва нужния за целта порт в сокета. Клиент, който желае да реализира комуникация със сървър, но не познава общо известните портове, трябва да притежава допълнителен механизъм за определяне към кой точно да бъде отправена заявката. Такъв механизъм може да се реализира посредством регистрирана карта на комуникационните портове, използваща общо познати такива.

Както беше изяснено формирането на комуникационна мрежа чрез свързването на съвкупност от мрежи (повече от две) е възможно чрез маршрутизатор (рутер). Тук е необходимо да се разбере разликата между различните типове маршрутизатори – рутери (routers), мостове (bridges) и шлюзове (gateways).

Мостове (bridge) – Тези устройства свързват отделни сегменти на локални компютърни мрежи на ниво мрежов интерфейс. Всеки bridge изпълнява функция за пренасочване на кадрите от подниво MAC (Media Access Control) и е независим от протоколите на по-високо йерархично ниво, включително и подниво LLC (Logical Link Control). Всеки bridge може да бъде прозрачен за IP. Това се получава в случаите, когато даден мост използва IP като комуникационен протокол, за да комуникира с друг, намиращ се на мрежов сегмент, свързан с bridge посредством изпращане на дейтаграми.

Рутери (router) – Устройствата свързват отделни сегменти на мрежово ниво и маршрутизират отделни пакети между тях. Всеки рутер трябва да може да разбере адресната структура, асоциирана с мрежовия протокол, да поддържа и взема решения как да маршрутизира съответния пакет или да го пренасочва.

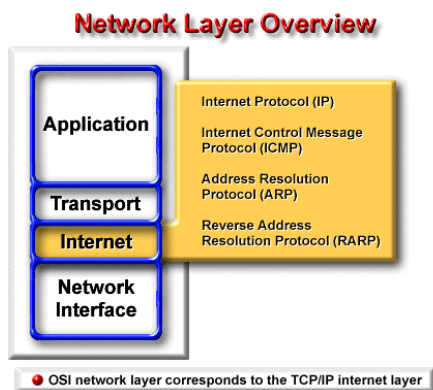
Рутерите имат възможност да определят най-добрия път за трансфер на информация и оптималната големина на пакета. Базовите рутиращи функции са реализирани в IP протокола. Съвременните рутери могат да изпълняват доста по-сложни функции от тези, които са дефинирани в IP.

Шлюзове (gateways) – Понятието gateway много често се смесва с входно/изходната точка при IP рутиране. В много случаи понятието gateway трябва да се употребява за реализация на входно/изходна точка на по-високо ниво за комуникация от рутер. Понякога такива устройства се използват за адресно преобразуване между отделни комуникационни мрежи. Използването на такива входно/изходни точки обикновено ограничава броя на конекциите между приложения, използващи ги за комуникация. Gateway е непрозрачен за IP комуникации. Ако определен хост изпрати дейтаграма през gateway, обикновено се реализира комуникация само до него, но не и прозрачно през него. Много често с това понятие се свързва и защитата, реализирана посредством „огнена стена” (firewall). Тя дава възможност за частично или пълно ограничаване на достъпа от една мрежа или група мрежи към друга такава.

1.2 Мрежово ниво

Протоколният стек IPv4 е в основата на изграждането на глобалната мрежова среда INTERNET, както и за проектирането на IP-базирани ведомствени мрежови конфигурации - INTRANET мрежови среди.

Мрежовото ниво на протоколния стек TCP/IP включва протоколите, предствени на фиг.1.3:



Фиг. 1.3 Протоколи на мрежово ниво

IP (Internet Protocol) е дейтаграмен мрежов протокол, осигуряващ несвързано мрежово обслужване. IP е протокол, който осигурява маршрутизирането на дейтаграмите от адреса-източник до целевия адрес, без да носи отговорност за достоверността на преноса на информацията и отстраняването на възникнали при предаването грешки. Протоколът поддържа йерархична адресна логика и осигурява глобален обмен на еднопосочни съобщения (дейтаграми).

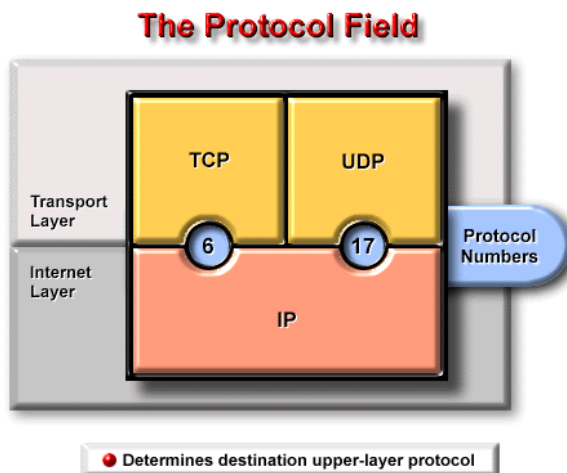
ICMP (Internet Control Message Protocol) е протокол за обмен на служебни съобщения на мрежово ниво. Използва се за управление и диагностика на състоянието на мрежовите съединения и обработване на аварийни ситуации.

ARP (Address Resolution Protocol) е протокол за намиране на адресно съответствие по валиден IP адрес. Той съпоставя IP (мрежовия адрес) на системата с нейния канален (физически) адрес (MAC-адрес).

RARP (Reverse Address Resolution Protocol) е протокол за намиране на адресно съответствие по валиден MAC адрес. Този протокол извежда съответствието между известен канален (физически) адрес (MAC-адрес) и присвоения на системата IP-адрес (мрежов адрес).

Протоколно взаимодействие

На фиг.1.6 е представено взаимодействието на базовия мрежов протокол IP и свързаните с него транспортни протоколи, осигуряващи свързано мрежово обслужване за приложните заявки.



фиг. 1.6 Номера на транспортни протоколи

Осигуряването на ориентирано към връзката мрежово обслужване се реализира в този протоколен стек на транспортно ниво, тъй като базовият мрежов протокол е дейтаграмен, т.е. неориентиран към връзката. Във взаимодействието участват два основни транспортни протокола:

TCP (Transmission Control Protocol) е ориентиран към връзката транспортен поротокол с две основни функции:

- да осъществява управление на предаването на информационни потоци като открива грешки, получени или неполучени пакети и осигурява тяхното повторно предаване;
- мултиплексиране на точките на достъп до мрежово обслужване при поддържането на приложните заявки.

Протоколът IP осигурява еднопосочното предаване на дейтаграми от адреса-източник до целевия адрес. Транспортният протокол TCP осигурява управлението на потока кадри и отстраняване на грешки.

На транспортното ниво в разглеждания протоколен стек, освен TCP, оперира и протокол за обмен на потребителски пакети *UDP (User Datagram Protocol)*. Този транспортен протокол е дейтаграмен, т.е. не предоставя механизъм за установяване на връзка и се използва за услуги, свързани с управление на мрежовите устройства и обмен на служебни съобщения.

Протокол ICMP

На мрежово ниво в разглеждания протоколен стек функционира протокола ICMP (Internet Control Message Protocol)

Протоколните съобщения на ICMP се пренасят в DATA-полето на IP-дейтаграмите и се използват за обмен на информация за грешки и управляваща информация.

Протоколът използва множество от командни примитиви, които дават информация за състоянието на мрежовото съединение:

- **Destination Unreachable** (недостижим IP адрес);
- **Time to Live Exceeded** (изтекло време на “живот”);
- **Parameter Problem** (проблеми при съставянето на дейтаграмите);
- **Redirect** (пренасочване на дейтаграми);
- **Echo** (заявка тест на IP адрес);
- **Echo Reply** (потвърждение за тест на IP адрес);
- **Timestamp** (заявка за времеви маркер);
- **Timestamp Reply** (потвърждение на времеви маркер)
- **Information Request** (заявка за информация за състояние);
- **Information Reply** (потвърждение за информация за състояние);
- **Address Request** (заявка за адресна информация);
- **Address Reply** (потвърждение за адресна информация).

Тази служебна информация се използва от приложенията за целите на анализ на състоянието на мрежовото съединение и обработване на ситуации на отказ.

Протоколи ARP и RARP

Един от основните проблеми при реализирането на мрежови съединения между компютри, включени в локални мрежови конфигурации, е проблема за взаимодействието между мрежови и физически адреси. Глобалният пренос се осъществява по мрежов адрес, докато мрежовия пакет, се транспортира в локалната мрежа в съответствие с конкретния протокол от канално ниво.

ARP се използва за построяване на необходимата за мрежовото съединение таблица на съответствието между присвоените на компютрите от локалната мрежа IP-адреси и MAC-адресите. Тази таблица позволява установяването на мрежови съединения, основани на преносната среда в локалната мрежа, например IEEE 802.3 Ethernet. При известен целеви IP адрес, за установяване на съединението е необходимо да се извлече от ARP-таблицата, съответстващия му MAC-адрес. Ако в ARP таблицата няма информация за това съответствие, то се изпраща IP-дейтаграма с целеви адрес, търсения целеви адрес, която се пакетира в broadcast MAC-кадър. Този кадър достига до всички MAC-адреси в локалната мрежа, т.е достига и до компютъра, чийто IP адрес е

целевия IP адрес. Компютърът потвърждава дейтаграмата, като на канално ниво се формира кадър с физически адрес на получател, адреса на инициатора на мрежовото съединение и канален адрес-източник – търсения MAC-адрес. След пристигане на потвърждението в инициращия компютър, неговата ARP-таблица се актуализира с липсващия MAC-адрес и са осигурени условията за установяване на мрежово съединение в MAC-преносната среда.

Протоколът RARP (Reverse Address Resolution Protocol) е “обратен” по функционалност на ARP. При този протокол се решава обратната задача, по известен MAC-адрес да се получи информация за съответстващия му IP-адрес. Използва се аналогичен подход, като се изпраща broadcast IP-дейтаграма, която достига мрежовото ниво и в потвърждението се съдържа търсения IP адрес.

1.3 Транспортни протоколи

Транспортното ниво се явява ниво за синхронизация, докато мрежовото ниво е нивото на информационния пренос, определен от ограниченията на комуникационната подсистема.

Транспортното ниво поддържа приложните заявки за обмен на информация и в средата на ограниченията на мрежовото ниво, осигурява заявеното качество на обслужване на приложните процеси. Транспортното ниво има за задача да осъществи прозрачен обмен на данни между сеансовите обекти и да ги освободи от изпълнението на функции по организиране на ефективното и надеждно предаване на данни.

Транспортният слой определя параметрите на качеството на мрежовото обслужване при предаването на данни, производителността на мрежата, подходящо мултиплексиране на мрежовите съединения и др. Тези параметри отчитат заявеното от обектите на сеансовото ниво качество на обслужване, от една страна, и от друга, отчитат реалните характеристики и възможности на комутационната подсистема, управлявана от мрежовия слой.

Функционалността на транспортното ниво е свързана с:

- мултиплексиране на точките на достъп до мрежовото съединение – установяване на няколко транспортни съединения на база на едно мрежово съединение;
- контрол на качеството на обслужване – следи за достоверното предаване на информацията, управлението на потока кадри, буфериране, откриване и корекция на грешки, контрол на времевите характеристики на информационния пренос и запълването на пропускателната способност на мрежовите съединения;
- установяване, поддържане и разпадане на транспортните съединения.

Транспортното ниво има синхронизиращи функции при използването на услугите на мрежовите съединения при зададено качество на обмена, зададено от сеансовото ниво.

“Високите” нива са част от операционната система, под управлението на която работи компютъра, или са приложение за операционната система, докато транспортното ниво е относително независимо от локалната операционна система, тъй като при него функционалността се определя от комуникационната

подсистема (първите три нива от еталонния модел). Транспортното ниво се организира в локалната архитектура като самостоятелен елемент в състава на мрежовото системно програмно осигуряване.

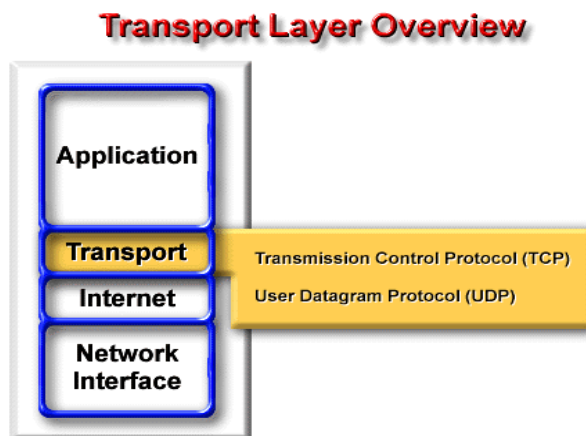
От архитектурен аспект транспортното ниво е първото от нивата, което реализира съединение точка – точка (Point to Point), от гледна точка на приложенията.

Съединението между приложните процеси се реализира на базата на транспортното ниво. На приложенията се предоставя за използване виртуално съединение, като комуникацията в “ниските” нива остава прозрачна. Независимо от разстоянието между двата крайни компютъра, функционалността на транспортния протокол е една и съща.

Всички приложно-ориентирани нива от еталонния модел се основават на транспортното, като достоверна комуникационна среда.

Транспортното ниво трябва да осигурява баланса между изискванията за комуникационни услуги на приложните нива и реалните възможности на мрежата.

В TCP/IP - протоколния стек, транспортният слой е изграден от два протокола TCP (Transport Control Protocol) и UDP (User Datagram Protocol) – фиг.1.7.



фиг. 1.7 Транспортни протоколи

1.3.1 Транспортен протокол TCP (Transport Control Protocol)

Интерфейсът между приложните програми (ПП) и TCP/IP услугата за надеждна доставка може да се характеризира с 5 основни свойства:

- **Ориентираност към потока** - когато две ПП (потребителски процеси) си обменят големи обеми данни, данните се разглеждат като потоци от битове, разпределени в 8-битови октети, наричани обикновено байтове. Услугата за доставка на поток от страна на получаващата машина предава на получателя точно същата последователност от октети, които подателят е предал на предаващата машина;

- **Виртуална връзка** - осъществяването на пренос на поток е

аналогично на осигуряването на телефонен разговор. Преди да започне преноса, двете ПП се свързват със своите операционни системи (ОС) и ги информират за желанието си да осъществят пренос на поток. По принцип, едната ПП прави "повикване", което трябва да бъде прието от другата ПП. Софтуерът на протоколите в двете ОС се свързват чрез обмен на съобщения през мрежата, като проверяват и дават разрешение (authorization), след което двете страни са готови. Щом подробностите бъдат уточнени, софтуерът на протокола информира ПП, че връзката е осъществена и следователно преносът може да започне. По време на преноса протоколният софтуер на двете машини продължава да поддържа връзка помежду им и проверява за коректното пристигане на данните. Ако по някаква причина връзката отпадне (напр. повреда в хардуера някъде по маршрута), двете машини отчитат това и го съобщават на съответните ПП. В такава ситуация, в рамките на изложението ще се използва термина виртуална верига за тези връзки, защото въпреки че ПП виждат връзката като налична хардуерна връзка, тя е виртуална връзка, предоставена от услугата за надеждна доставка;

- **Буфериран пренос** - ПП изпращат потоци от данни по виртуалната верига чрез непрекъснато изпращане на октети към протоколния софтуер. При предаването на данни всяка ПП използва произволни по големина сегменти от данни, които могат да съдържат и само един октет. След проверка за коректност от страна на получателя, протоколният софтуер незабавно предава на получаващата ПП октетите точно в същия ред, в който те са били изпратени. Протоколният софтуер може да разделя потока в пакети с размери, нямщи нищо общо с получаваните от ПП парчета. За повишаване на ефективността и намаляване на мрежовия трафик, различните реализации на протокола за пренос обикновено набират от потока достатъчно данни до запълването на разумна по големина дейтаграма, която изпращат по мрежата. Така, даже ПП да изпраща периодично по един октет, преносът по мрежата може да бъде достатъчно ефективен. По същия начин, ако ПП реши да генерира много голям блок от данни, протоколният софтуер го разделя на по-малки части преди да го предаде по мрежата. За тези приложения, при които данните трябва да се пренесат, даже и да са много малко, услугата осигурява *push* механизъм, който ПП използват, за да предизвикат незабавен трансфер. От страна на подателя, това принуждава протокола да изпрати всички генерирани данни без да изчаква запълването на буфера. Когато тези данни пристигнат при получаващата машина, *push* принуждава ТСП да предаде данните на получателя без забавяне. Трябва да се има предвид, че *push* механизмът гарантира само преноса на всички данни, но няма ограничение за обема. Така че, даже при такова принудително пренасяне, протоколът пак може да раздели потока по избран от него начин;

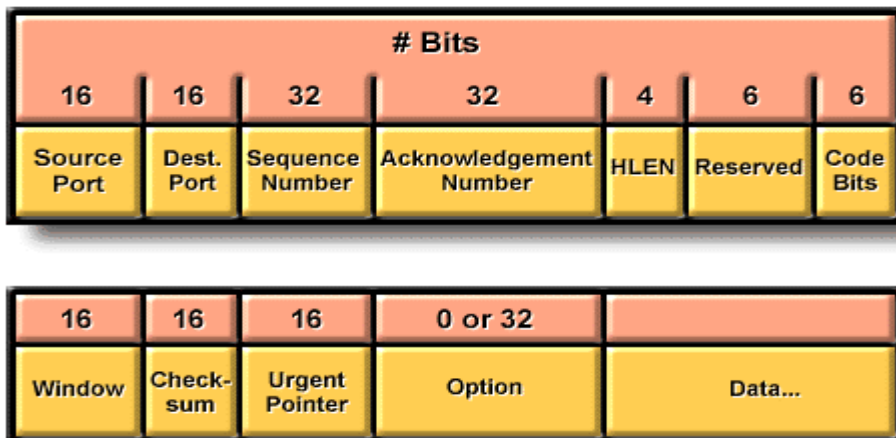
- **Неструктуриран поток** - важно е да се разбере, че ТСП/IP услугата за пренос на поток не различава структурираните данни. Не е предвиден начин програма за изплащане на заплати да принуди услугата да постави граници между записите за отделни служители, или да идентифицира съдържанието като ведомост за изплащане на заплати. Приложните програми са тези, които

трябва да разбират съдържанието на потока и да договорят формата на данните преди да иницират връзка;

- **Пълно-дуплексна връзка** - връзките, осигурявани от TCP/IP услугата за пренос на поток, дават възможност за едновременен пренос в двете посоки. Такива връзки се наричат пълно-дуплексни. От гледна точка на приложния процес, пълно-дуплексната връзка се състои от два напълно независими потока, течащи в противоположни направления, без явна връзка между тях. Пренасящата услуга позволява на даден приложен процес да спре потока по едното направление, докато данните продължават да се пренасят по другото, при което връзката става полу-дуплексна. Предимството на пълно-дуплексната връзка е, че протоколният софтуер може да изпраща контролираща информация за единия поток обратно до източника в рамките на потока дейтаграми, пренасяни в обратното направление. Това намалява мрежовия трафик.

На фиг.1.8 е представена формата на TCP-протоколната спецификация.

The TCP Segment Format



фиг. 1.8 Структура на заглавната част на TCP сегмент

Приложните услуги, основани на базата на IP-протоколния стек, са представени на фиг. 1.9.

ПРИЛОЖНИ УСЛУГИ						
HTTP	SMTP	FTP	TELNET	SNMP	UDTP	и т.н.
<u>TCP</u>				<u>UDP</u>		
<u>Internet IP v 4.0</u>						

Фиг. 1.9 Услуги на приложно ниво

Една част от услугите изискват ориентирано към връзката мрежово съединение и се поддържат на транспортно ниво от протокол TCP:

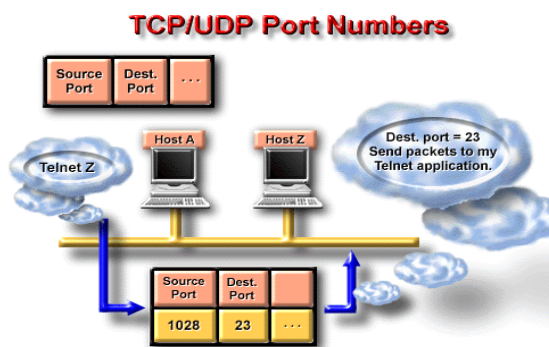
- World Wide Web – HTTP (Hiper Text Transfer Protocol) протокол;
- Електронна поща (MAIL) – поддържа се от протокол SMTP (Simple Message Transport Protocol);
- Обмен на файлове – FTP (File Transfer Protocol) протокол.

Втора група приложни услуги не поставят изискването на свързано мрежово обслужване. Тогава на транспортно ниво се използва протокола UDP:

- Отдалечено администриране на устройства – SNMP (Simple Network Management Protocol) протокол;
- UDTP (User Define Transport Protocol) – този протокол позволява допълнително дефиниране на протоколна спецификация на приложно ниво.

Някои от портовете са резервирани от двата протокола и приложенията не могат да използват тези портове. Принципът на разпределение на портовете е следния:

- портове с номера до 255 са предоставени за общо използване;
- портове с номера от 256 до 1023 са за системни приложения;
- портове с номера над 1023 са недефинирани.

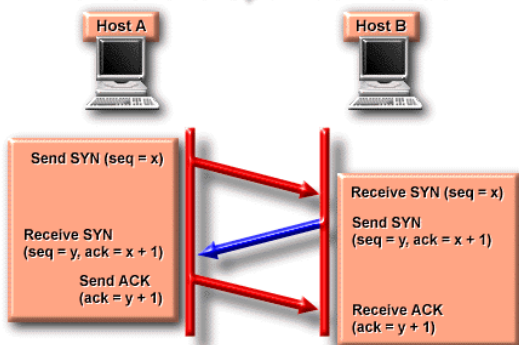


фиг. 1.11 Разпределение на портове в процес на връзка

Крайните компютри използват портовете, за да достигнат до желаното приложение, като в полето за порт на получател (Dest. port) се указва номера на порта, с който се адресира на транспортно ниво приложението – номер 23 – приложение TELNET – фиг. 1.11.

Установяването на TCP съединението в двете крайни точки (компютри) се реализира на три стъпки по последователността, показана на фиг. 1.12.

The TCP Three-Way Handshake/Open Connection

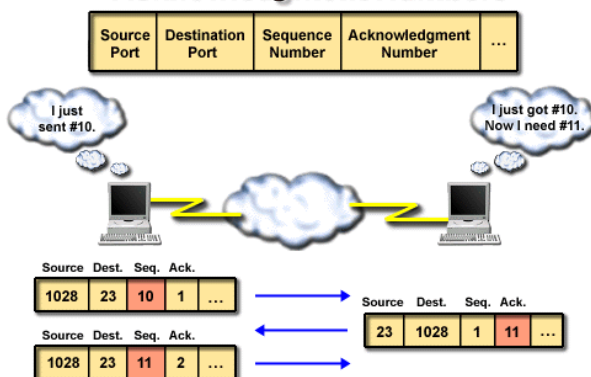


Фиг. 1.12 „Трипътно ръкостискане” при TCP протокол

Процедурата е фактически тест на логиката за потвърждение на протокола, като се обменя един празен сегмент в двете посоки и се изчаква двупосочно потвърждение за него преди да стартира обмена на реална информация.

Една примерна процедурна последователност за дуплексно управление на потока сегменти при TCP е представена на фиг.1.14

TCP Sequence and Acknowledgment Numbers



фиг. 1.14 Последователни стъпки на връзката (трипътно ръкостискане) при TCP протокол

В полето за номер на сегмента се записва номера на текущо изпратения сегмент, а в полето за потвърждение се указва номера на сегмента, който се очаква да бъде предаден, с което се потвърждава текущо предавания сегмент.

Проблем по отношение на надеждността може да възникне, когато доставящата система от по-нисък протоколен слой дублира пакет. Дублирани пакети възникват и когато в мрежата има големи закъснения, предизвикващи повторно изпращане на пакети. Решаването на този проблем изисква сериозен размисъл, тъй като е възможно да се дублират не само пакети, но и

потвърждаващи съобщения. Обикновено надеждните протоколи откриват дублираните пакети чрез присвояване на последователен номер на всеки пакет, като изискват от получателя да помни кои номера от последователността е получил. За избягване на объркването от закъснели или дублирани потвърждения, те също съдържат номера на потвърдения пакет.

Методът на "пълзящия прозорец" е по-сложен. Протоколите, които използват пълзящия прозорец", по-добре използват пропускателната способност на мрежата, защото дават възможност на подателя да изпраща множество пакети преди да изчака за тяхното потвърждение.

Даден пакет се нарича непотвърден, ако е бил изпратен, но все още не е получено потвърждение, че е получен. Броят на непотвърдените пакети технически е ограничен от размера на прозореца, който обикновено е някакво малко число.

За да бъде изяснено защо пропускателната способност остава висока при обикновени връзки, трябва да се има в предвид, че приложните програми, оптимизирани за високоскоростен пренос, не генерират данни октет по октет (ако го правят, то ще предизвикат нежелано забавяне в операционната система). Такива програми излъчват големи блокове данни с единични заявки. Тогава буферът на изпращащият ТСР започва с достатъчно данни поне за един сегмент с максимална дължина. По-нататък, тъй като програмата генерира данни по-бързо, отколкото ТСР може да ги пре-несе, буферът при подателя остава почти пълен и ТСР не забавя преноса. В резултат ТСР продължава да изпраща сегменти с максималната скорост, която позволява междумрежовата среда, докато приложната програма продължава да пълни буфера.

ТСР изисква и подателя, и получателя да използват методики за избягване на синдрома на тъпия прозорец. Получателят избягва да съобщава за малък размер на прозореца, а подателят използва адаптивна схема за забавяне на изпращането, за да групира данните в големи сегменти.

1.3.2 Транспортен протокол UDP (User Datagram Protocol)

В пакета протоколи ТСР/ІР този протокол осигурява първичния механизъм, който използват приложните програми за изпращане на дейтаграми до други приложни програми (ПП). UDP осигурява портове, които служат за различаване на множеството-програми, изпълнявани на съответната машина. Т.е., освен самите данни, UDP съобщението съдържа номерата на порта-получател и порта-подател, така че UDP софтуера да може да изпрати съобщението на правилния получател, който пък има възможност да върне отговор.

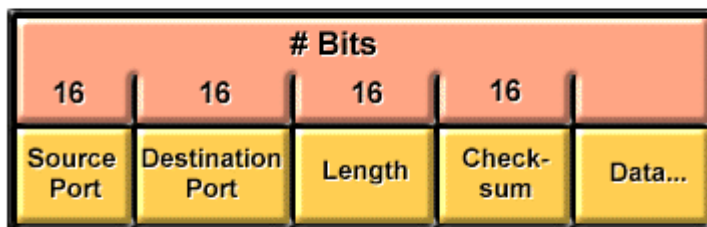
UDP използва намиращият се под него протокол ІР за транспортиране на съобщението от една машина до друга, като осигурява същата ненадеждна несвързана с потока данни семантика на изпращането на дейтаграмата, както ІР. Той не използва опознаване (acknowledge) за да проверява дали съобщенията пристигат, не подрежда идващите съобщения, не дава обратна връзка за контрол на скоростта, с която се предава информацията между двете машини. Така UDP съобщенията могат да се загубят, дублират или

да пристигат в непоследователен порядък. Още повече пакетите могат да пристигат с по-голяма скорост, отколкото получателя може да ги обработва. Протоколът UDP осигурява ненадеждна услуга с използване на IP за транспортиране на съобщенията между машините. Той използва IP като носител, но добавя способността да се различават множество получатели в рамките на един компютър.

ПП, която използва UDP, поема пълната отговорност за обработка и справяне с проблемите на надеждността, включително загуба на съобщения, дублиране, закъснение, разбъркан порядък и загуба на връзка. За съжаление, приложните програмисти често пренебрегват тези проблеми при проектирането на софтуер. При това, тъй като приложните програмисти често тестват мрежовия софтуер върху бързи, надеждни и ниски закъснения локални мрежи, тестовите може да не покажат слабостите и потенциалните проблеми. Така много ПП, които се опират на UDP, работят добре в локални мрежи, но пропадат в големи TCP/IP мрежи.

На фиг.1.15 е представен формата на UDP-сегмента

The UDP Segment Format



● No sequence or acknowledgement fields

фиг. 1.15 Формат на заглавна част в UDP дейтаграма

Сравнявайки този формат с формата на TCP сегмента се наблюдава отсъствие на полета за контрол на потока и корекция на грешки. Това е характерно за дейтаграмните протоколи. Във функционалността на сегмента е съхранена възможността за мултиплексиране на мрежовите съединения

Полетата SOURCE PORT (порт подател) и DESTINATION PORT (порт получател) съдържат в 16-битово представяне номерата на UDP портовете, които се използват за де мултиплексиране на дейтаграмите между процесите, които очакват да ги получат. Портът подател може да не се попълва. Ако това поле бъде използвано, то възможните отговори ще бъдат изпратени са указания номер на порт. Ако не се използва, стойността трябва да бъде 0.

Полето LENGTH (дължина) съдържа общия брой на октетите в дейтаграмата, включително заглавната част и данните. Следователно, минималната стойност в това поле ще бъде 8, колкото е броя на октетите в заглавната част.

Полето за контролната сума се използва по избор. Стойност 0 в това поле означава, че не е пресмятана контролна сума. Възможността да не се пресмята контролна сума е оставена за случаите, когато UDP се използва във високонадеждна локална мрежа и не е необходимо да се губи време за пресмятането и. Нека не забравяме, все пак, че IP не пресмята контролна сума за данните в дейтаграмата. Така че контролната сума на UDP дава единствената възможност за гарантиране на точността на данните, пристигащи при получателя, и би трябвало да се използва.

Много от потребителите често се чудят какво става с UDP съобщенията, за които е пресметната контролна сума е 0. Това е възможно, защото UDP използва същия алгоритъм за пресмятане на контролната сума, както IP: той разделя данните на 16-битови стойности и пресмята допълнението към допълнената сума от отделните битове. Оказва се, че нулата не е проблем, тъй като в тази "допълнителна" аритметика нулата може да се представи по два начина: всички битове са 0 или всички битове са 1. Когато пресметнатата контролна сума е 0, UDP използва представянето с единици.

Контролната сума на UDP покрива повече информация, отколкото се съдържа в UDP дейтаграмата. За пресмятането на контролната сума UDP наставя псевдо-заглавна част към дейтаграмата, добавя октет от нули за да формира дейтаграма от цяло число 16-битови фрагменти и пресмята контролна сума за целия получен обект. Добавените псевдо-заглавна част и октета от нули не се изпращат с дейтаграмата, нито пък се вземат предвид при пресмятането на дължината и. За пресмятането на контролна сума софтуерът най-напред записва 0 в полето за контролна сума, после прави пресмятането за целия обект заедно с псевдо-заглавната част, истинската заглавна част и данните.

Псевдо-заглавната част се използва за проверка дали UDP дейтаграмата е пристигнала при получателя. Важно е тук ясно да се разбира, че адресът на получателя се състои от адреса на машината и номера на порта в тази машина. Заглавната част на UDP определя само номера на порта. Тогава, за да се гарантира получаването, UDP в машината на подателя пресмята контролна сума, която обхваща UDP дейтаграмата и IP адреса на получателя. В машината на получателя UDP проверява контролната сума като използва IP адреса на получателя, записан в заглавната част на IP дейтаграмата, която пренася UDP съобщението. Ако контролните суми съвпадат, то дейтаграмата е пристигнала точно по местоназначението и на правилния порт.

При пристигането си, най-долният слой от мрежовия софтуер приема пакета и той започва да се изкачва по слоевете. Всеки слой премахва съответната заглавна част и предава остатъка на горния слой, така че най-горният слой предава данните на очакващия ги процес без заглавна част. Така най-външната заглавна част съответства на най-ниския мрежов протокол. При разглеждането на добавянето и премахването на заглавни части трябва да имаме предвид принципа на слоевете. В частност този принцип се прилага при UDP, така че UDP дейтаграмата, получена от IP на машината-получател, е идентична с дейтаграмата, която UDP е предал на IP в машината-подател. По същия начин данните, които UDP предава на потребителския процес в машината-получател,

ще бъдат точно данните, които друг процес е предал на UDP в машината-подател.

Разпределението на задълженията между различните слоеве протоколи е точно и ясно:

- IP слоят отговаря само за преноса на данни между двойка хостове през мрежата;
- UDP отговаря само за разпределянето между многото възможни податели и получатели в рамките на тези хостове.

По този начин само IP заглавната част идентифицира машините подател и получател; само UDP идентифицира портовете в рамките на един хост.

Тук възниква въпроса: Как се присвоява номер на порт?

Проблемът е важен, тъй като два компютъра трябва да съгласуват номерата на портовете си преди да започнат обмен.

Например, когато компютър А иска да получи файл от компютър В, той трябва да знае кой порт използва компютър В за трансфер на файлове. Има два фундаментални подхода при присвояването на номера на портове. При първия подход се използва централизирано определяне. Всички останали се съгласяват и позволяват на "централната власт" да дава номера на портове и да публикува техния списък. Тогава софтуерът се изгражда според този списък. Този подход се нарича понякога "универсално присвояване" и присвоените номера на портове се наричат "известни номера на портове".

Вторият подход използва динамично свързване. При него номерата не са глобално известни. Вместо това, когато дадена програма поиска порт, мрежовият софтуер го дава.

За да се види моментното състояние на портовете на друг компютър ще трябва да се изпрати заявка с въпрос от вида "Кой порт се използва за трансфер на файлове?" Отговорът съдържа номера на порта, който трябва да се използва в този случай.

Доколкото е възможно, други транспортни протоколи, които предлагат същите услуги, използват същите номера на портове.

Създателите на TCP/IP са използвали хибриден подход, при който някои от портовете имат предварително зададени номера, а останалите се управляват динамично от локалните ОС или ПП. Предварително зададените и еднакви за всички номера използват ниски стойности, като по-високите номера са оставени за динамично присвояване.

Обобщение

Протоколът за контрол на преноса TCP дефинира ключова услуга в междумрежовата среда, а именно надеждния пренос на поток от данни. TCP осигурява пълнодуплексна връзка между две машини, като им дава възможност да обменят ефективно големи обеми данни.

Тъй като използва протокол за пълзящ прозорец, TCP може ефективно да използва мрежата. Освен това, той малко зависи от разположената под него системи за доставка и поради това е достатъчно гъвкав, за да може да работи с

голямо разнообразие от системи за доставка. Тъй като осигурява контрол на потока, TSP дава възможност за комуникация между системи с много различни бързодействия.

Основната единица, в която TSP пренася данни, е сегмент. Сегментите се използват за обмен на данни и контролна информация (например, за да даде възможност на TSP софтуера на две машини да установява връзка и да я прекратява). Форматът на сегмента позволява на една машина да вмъкне потвържденията за данни пристигнали от едната посока в заглавните части на сегменти с данни, изпратени в обратната посока.

TSP осъществява контрол на потока посредством съобщение от страна на получателя за това колко данни може да приеме. Освен това той поддържа и съобщения извън пренасяния поток чрез специално средство за спешно изпращане на данни и предизвиква доставянето чрез механизъм push.

Съвременния стандарт за TSP дефинира експоненциално увеличаване на таймерите за повторно изпращане и алгоритми за избягване на задръстване, ката бавен старт, степенно намаляване и стъпково нарастване. В допълнение, TSP използва специални методики за избягване на изпращането на малки пакети

Глава втора

Заплахи за сигурността на мрежата

2.1 Мрежова сигурност

Мрежовата сигурност е сериозен проблем в света на информационните технологии. Мрежите на редица организации постоянно са атакувани от широко разпространени атаки чрез компютърни вируси както и от квалифицирани компютърни хакери. От друга страна администраторите на множество корпоративни мрежи и потребители на Интернет, т.е почти всеки, който е „онлайн“, е заинтересован до някаква степен от проследяването на възможността за неоторизиран достъп.

За различните хора сигурността означава различни неща. Компютърната сигурност може да бъде дефинирана като съвкупност от „необходимите действия за защита на един компютър и информацията, която той съдържа“. Все пак трябва да се добави, че дори и след това няма никакви гаранции за така реализираната „сигурност“.

Както е добре известно всички потребители предпочитат да имат по-свободен достъп, а администраторите - да го ограничават с цел постигане на по-голяма сигурност. Това може да доведе до създаване на неприятелски отношения между мрежовия администратор и потребителите в мрежата..

Именно затова е необходимо още от самото начало да се дефинира желаната сигурност на мрежата. Поставя се следния въпрос:

Каква сигурност ще бъде достатъчна за желана компютърна мрежа?

Отговорът зависи от организацията, в която ще се изгражда мрежата. Първата стъпка при разработването на даден план за защита на компютърните данни е определянето на нивото на желаната сигурност. Факторите, които трябва да бъдат оценени, са следните:

- Типът на бизнеса, в който участва компанията;
- Типът на данните, съхранявани в мрежата;
- Философията за управлението на организацията.

Тип на бизнеса

Някои типове бизнес като юридически и медицински услуги генерират поверителни данни. Частният характер на медицинските данни на пациента или комуникацията адвокат-клиент са защитени от закона. Ако в мрежата се съхраняват конфиденциални документи, определено трябва да бъде поддържано високо ниво на сигурност. В противен случай организацията е подложена на риск от наказателно преследване.

Редица организации често използват поверителни данни:

- Агенциите на изпълнителната власт, съдилищата и други правителствени институции;
- Образователни институции и болници;
- Компании, които са свързани с националната сигурност;

- Всяка организация, която събира данни срещу гаранция за конфиденциалност;
- Всяка организация, която произвежда продукт или предоставя услуга в областта на индустрията с висока конкурентност, или прави научни изследвания в такава област;
- Всяка организация, мрежата на която е свързана към Интернет;

Тип на данните

Независимо от типа на бизнеса, определени типове данни се считат за частни и трябва да бъдат защитени. Такъв тип данни са:

- Заплатите и персоналната информация на служителите
- Счетоводната и данъчната информация
- Търговски тайни, като оригинален код, планове и диаграми, рецепти и бизнес стратегии

Ако в мрежата се съхранява такъв тип информация, трябва да се реализира план за сигурност с цел защита на данните.

Философия на управление

Често данните в мрежата не са от частен характер. Тогава нивото на сигурност може да зависи от персоналната философия на собствениците или мениджърите на бизнес организацията, като се вземе предвид колко отворена (или затворена) искат да бъде мрежата.

Във всяка една организация достъпността и лекотата от използването на информация в мрежата, се радват на по-голяма популярност, отколкото конфиденциалността и сигурността. В редица организации се действа според принципа „необходимо му е да знае“, като в случая информацията трябва да бъде достъпна само за този, чиято работа я изисква.

И двете политики не са напълно коректни. Мрежовите администратори трябва да познават и реализират сигурността на мрежата съобразно стила на управление на организацията.

Определен тип атаки като разпределен отказ на услуга (Distributed Denial of Service, DDoS) атакуват работата на мрежата, като използват компютри посредници и подмолно инсталират софтуер на тях, който ще бъде използван като част от атакуващата платформа. По този начин компютрите на компанията може неволно да участват в атаката и в резултат от това да си навлекат наказателна отговорност. По тази причина е важно всички мрежи, свързани към Интернет, да реализират политики на сигурност независимо от степента на конфиденциалност на техните собствени данни.

2.2 Видове заплахи за сигурността

На основата на типа на бизнеса, данните и философията на управление на фирмата се изисква реализиране на определени мерки за сигурност. Това означава, че трябва да се определят вероятните източници на заплаха за целостта на данните.

Организациите могат лесно да подценят, надценят или напълно да пренебрегнат рисковете, на които са изложени техните мрежи. Въпреки това има много различни типове заплахи за сигурността на мрежата.

Тези заплахи възприемат много различни форми, но всички те водят до загуба на поверителност до някаква степен и вероятно злонамерено разрушаване на информация или ресурси, което може да доведе до големи парични загуби.

Информацията за това кои области от мрежата са по-податливи за атаки и кой е най-честият атакуващ е изключително полезна.

Преди време се вярваше на потребители, които са ползватели на вътрешната корпоративна мрежа. Считаше се, че не трябва да се вярва на връзки, произлизащи от Интернет или от мрежи за отдалечен достъп посредством виртуални частни мрежи (VPNs), dial-in. Доверие се имаше на служителите, които са вътрешни за мрежата, и на оторизирани хора, опитващи се да използват вътрешни мрежови ресурси, извън корпорацията.

Нещата сега са коренно различни и доверието трябва да съответства на реалността.

По настоящем повече от атаките се предприемат от вътрешни хора на фирмата, и вече има засилваща се тенденция да не се вярва на вътрешни потребители и да се прилагат съответни по-строги мерки за сигурност. От друга страна безжичните мрежи стават все по-широко разпространени и често в тези случаи се изискват по-строги приготовления. Ограниченото използване на оборудване от мрежовата инфраструктура и критичните ресурси е необходимо. Ограничаването на достъпа само до онези, които изискват достъп, е умен начин за възпиране на много заплахи, пробиващи сигурността на компютърните мрежи.

Всичко това показва, че трябва да бъдат отчетени редица фактори и заплахи за сигурността на една мрежа.

От друга страна не всички заплахи са замислени като злосторни, но те могат да покажат същото поведение и могат да причинят също толкова щети - независимо дали преднамерено или не. За съжаление, много мрежови инфраструктури трябва да се справят с нарастващите проблеми с вируси и малуер, които могат да бъдат открити върху компрометирани изчислителни ресурси и поставят несъзнателни заплахи за сигурността от страна на нищо-неподозиращи служители.

Затова е важно да се разбере какви видове атаки и слабости са често срещани и какво можете да се предприеме на ниво политика, за да се гарантира някакво ниво на сигурна мрежа.

Заплахите могат да бъдат класифицирани следните две категории:

- Външни заплахи
- Вътрешни заплахи

Външни заплахи

Преди време мрежата беше самостоятелна и външните заплахи не бяха сериозен проблем за локалните мрежи (LAN). Сега повечето LAN мрежи са

свързани към Интернет. Това означава, че когато мрежата осъществява достъп до външния свят, външните хора също имат достъп до мрежата.

Мотивите на външните нарушители са различни. Най-често срещаните мотиви са отмъщение (недоволни клиенти, вечно обидени бивши служители и сърдити конкуренти), развлечение (такива, които се „хакват“ в мрежите „просто за забавление“, или за да докажат своите технически умения) и срещу заплащане. В последния случай на нарушителя е заплатено да нахлуе в мрежата, или той прави това за лична изгода; например хакерът може да се опита да прехвърли парични суми на своя собствена банкова сметка или да изтрие записите за своите дългове.

Външните пробиви в сигурността могат да приемат множество форми като:

- Неоторизирано използване на пароли и ключове;
- Атаки от типа Denial of Service (DOS);
- IP спуфинг;
- Компютърни вируси и червеи;
- Програми тип троянски коне.

Вътрешни заплахи

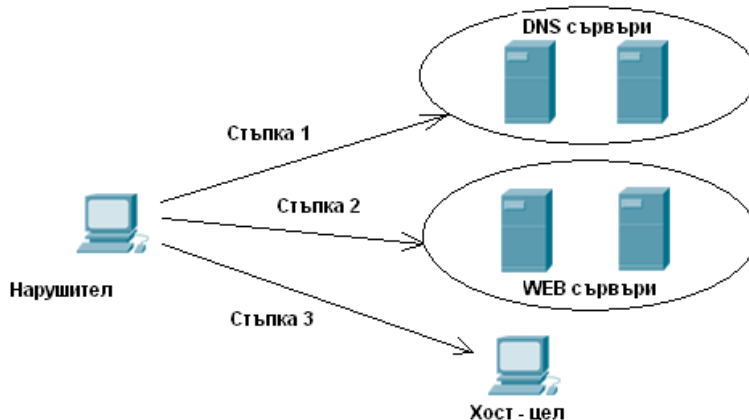
Много политики на сигурност в мрежата се фокусират само върху Интернет и външните заплахи. Това обикновено е грешка. Мрежовите администратори не трябва да пренебрегват риска от вътрешни пробиви на сигурността. Много случаи на кражба, злоупотреба или унищожаване на данни са „вътрешна работа“.

Съществуват няколко мотива за вътрешни пробиви на сигурността::

- Корпоративен шпионаж
- Вътрешни политики
- Недоволни служители (включително бивши служители)
- Случайни пробиви

Сканиране на портове

Когато бъдат открити работещи системи, атакуващият обикновено се опитва да открие кои услуги са на разположение за експлоатация. Това може да се постигне чрез техника, известна като **сканиране на портовете**. Както се знае всяко приложение е с присвоен номер на порт, който го идентифицира. Използвайки скенери на портове, нарушителите имат възможност да се сдобият с достъп до информация за това, кои приложения и мрежови услуги са на разположение за експлоатация. На фиг. 2.1 показан пример за проучване на достъпен хост.



Фиг. 2.1 Проучване за достъпност на хост

Нарушителят може да изпълни няколко стъпки, за да се сдобие с неоторизиран достъп до даден уеб сървър:

Стъпка 1 - DNS заявка, за да определи кои уеб сървъри са на разположение.

Стъпка 2 - Командата *ping*, за да определи кои сървъри са работещи и достъпни.

Стъпка 3 - Сканиране на портовете, за да определи кои услуги са на разположение за експлоатация.

Мрежовото разузнаване не може да бъде предотвратено изцяло. Ако Internet Control Message Protocol (ICMP) ехото и отговорът на ехо бъдат изключени за крайните маршрутизатори, *ping* пробите могат да бъдат спрени, но за сметка на данни за мрежова диагностика. Сканиранията на портове обаче могат да се провеждат с лекота без предварително използване на *ping*; те просто отнемат повече време, защото трябва да сканират IP адреси, които може и да не са активни.

Системите за засичане на нарушения (IDS) на нива мрежа и хост обикновено са в състояние да известят администратора, когато започне разузнавателна атака. Това му дава възможност да се подготви по-добре за приближаващата атака или да извести Интернет доставчика (ISP), хостващ системата, която стартира разузнавателното сондиране.

Подслушване на физически кабел

Дали е лесно или трудно на дебненето на пакети (известно и като *подслушване*) по мрежите, зависи силно от приложената технология. Мрежите с общи среди са особено податливи на подслушване, защото този вид мрежи предават пакети навсякъде по мрежата при пътуването им от източника до крайната цел.

Когато в някоя среда с обща медия се използват концентратори, или хъбове, (например FDDI, 10BASE-T или 100-Mbps Ethernet), може да бъде относително лесно да се добави нов възел с възможност за улавяне на пакети и след това да се подслушва трафика по мрежата.

Както е показано на фигура 2.2, атакуващият има възможност да се включи към Ethernet комутатор и чрез програма за декодиране на пакети, например Wireshark или TCPDump, да чете данните в Ethernet мрежата.



Фиг. 2.2 Неоторизиран достъп посредством *Wireshark/TCPDump*

В случая нарушителят получава достъп до информация за потребителски имена/пароли и поверителна информация от протокола за маршрутизиране посредством декодер на Ethernet пакети от рода на *Wireshark*. Изпращаните пакети с данни се прихващат от преносимия компютър, на който работи *Wireshark*; програмата декодира шестнадесетичните данни във вид, разбираем за хора. След като се слобие с достъп до информацията, нарушителят може да използва тази информация, за да получи достъп до някоя машина, както и евентуално до забранена за копиране лична информация и програми. Той има възможност, също така, в последствие да промени ресурса, т.е. нарушителят може да променя записи в сървъра или да променя съдържанието на информацията за маршрутизиране.

Фигура 2.3 показва пример за комутатор, притежаващ възможността да научава MAC адреси и да предоставя някаква сигурност на портовете.



Фиг.2.3 Сигурност на портове при *Ethernet* комутатори

Комутаторът 100BASE-T Ethernet предоставя свързаност на няколко хоста. Той научава MAC адреса на източника от комуникаращите си хостове и поддържа вътрешна таблица (Content Address Memory, CAM), представляваща MAC адреса и асоциираните портове. Когато някой порт получи пакет, комутаторът сравнява адреса на източника на този пакет с адреса на източника, научен от порта. Когато настъпи промяна в адреса на източника, се изпраща известие до контролна станция и портът може да бъде автоматично блокиран, докато се разреши конфликта.

Безжичен достъп

Безжичните мрежи са особено податливи на неоторизиран достъп. Безжичните точки за достъп се внедряват широко в корпоративните мрежи, защото те лесно разширяват свързаността с корпоративните потребители без да се губи време и пари за окабеляване. Тези безжични точки за достъп (Access Point, AP) служат за мостове и разширяват мрежата с още около 150 метра. Много летища, хотели и дори кафенета дават безплатен безжичен достъп, и поради това почти всеки с безжична карта на своето мобилно устройство е оторизиран потребител. Безжичните мрежи позволяват достъп и администраторите може въобще да нямат представа колко лесно някой може да получи достъп до тези мрежи. Броят на безжичните мрежи без включени никакви мерки за сигурност е изключително голям. Повечето хора пускат своите AP да работят в отворен режим, което означава, че в общи линии са широко отворени и няма включено никакво криптиране. Голяма част от тях използват подразбиращи се Service Set Identifier (SSID) и IP области, което означава, че при конфигурирането на безжичните мрежи не е правена никаква, или почти никаква, настройка.

Безжичните мрежи от стандарт 802.11 и съответните карти и точките за достъп на пазара прилагат стандарт за безжично криптиране, наречен Wired Equivalent Protocol (WEP), който на теория затруднява достъпа до безжичната мрежа без оторизация или пасивното подслушване на комуникациите. WEP обаче е с много вродени слабости, даващи възможност на нарушителите да пробият криптирането посредством сложен софтуер и стандартно оборудване. Затова е по-правилно да се използват по-добрите варианти за криптиране като Temporal Key Integrity Protocol (TKIP), Light Extensible Authentication Protocol (LEAP), Protected Extensible Authentication Protocol (PEAP) и т.н. Необходимо е всички настройки по подразбиране да бъдат променени, така че да се активират разумни услуги за удостоверяване и поверителност. Това ще направи голяма крачка към намаляването на неоторизирания достъп.

Фигура 2.4 показва пример за нарушител, получаващ достъп до безжична мрежа. Без значение кой метод е използван за началния неоторизиран достъп - разузнавателна работа, достъп през Интернет, подслушване на физически кабел, отдалечен достъп през dial-in модем или достъп през безжична мрежа - най-добрият начин за възпиране на неоторизирания достъп е чрез използването на услуги за поверителност и цялост, за да е сигурно, че трафикът, преминаващ през канала, е нечетлив и не може да бъде променян по време на преноса.

Таблица 2.1 изброява някои често срещани пропуски в достъпа и как те стават заплаха за корпоративните мрежи.



Фиг. 2.4 Получаване на неоторизиран достъп до безжична мрежа

Атаките с измама (IP спуфинг, Spoofing) включват предоставянето на фалшива информация за идентичността на някой абонат на мрежа, с цел да се получи неоторизиран достъп до системи и техните услуги.

IP спуфингът (подправянето) включва промяна на хедърите на пакети на изпращаните съобщения. Това ги кара да изглеждат така, сякаш идват от IP адрес, различен от реалния адрес на първоизточника. Макар че спуфингът само по себе си не е форма на атака, той представлява метод за придобиване на неоторизиран достъп до компютър или мрежа за започване на атака, за кражба на данни или за унищожаване на данни.

Атаката с повторение (replay) може да бъде вид атака с измама, защото съобщенията се записват и по-късно се изпращат отново, обикновено за да се използват пролуки в схемите за удостоверяване. Както атаките с измама, така и атаки с повторение обикновено следват в резултат на подслушване. Много програми за подслушване на пакети притежават и възможности за генериране на пакети, като могат да прихващат пакети с данни и по-късно да ги възпроизвеждат отново.

Въплъщаването в други хора е обичайно. Повечето от тези сценарии спадат към получаването на достъп до поредици за удостоверяване и след това използване на тази информация за получаване на неоторизиран достъп. След като достъпът е осигурен, причинените щети зависят от мотивите на нарушителя.

С помощта на механизмите за криптографско удостоверяване, атаките с въплъщаване могат да се предотвратят. Допълнителна полза от тези механизми за удостоверяване е това, че в някои случаи се постига и неот-ричане. Някой от потребителите, участващи в дадена електронна комуникационна обмяна, няма възможност в последствие да се отрече от изпращането на дадено съобщение. Това потвърждаване е критично за ситуациите, включващи електронни финансови операции или електронни договори, защото това са областите, в които хората най-често отричат участието си в незаконни действия.

Въплъщаването в устройство до голяма степен се състои в изпращането на пакети с данни, които се считат за валидни, но които може да са измамни. Обикновено тази атака предизвиква нежелателно поведение в мрежата. Въплъщавайки се в маршрутизатор и изпращайки модифицирана информация за маршрутизиране, нарушителят може да постигне по-добра свързаност за определен потребител.

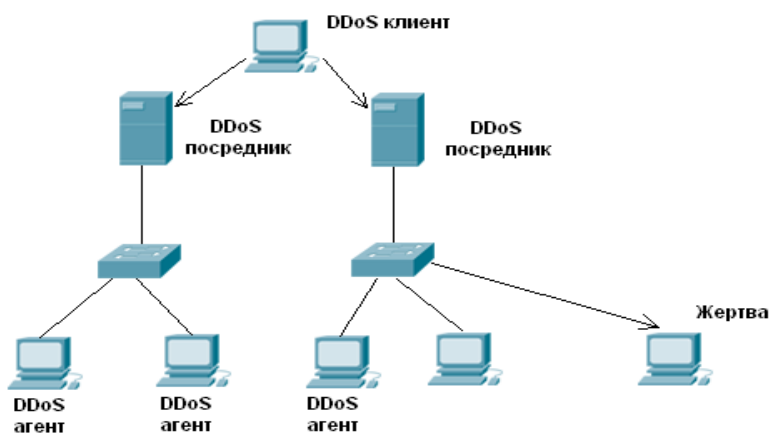
С прихващането на информацията за маршрутизиране и с достатъчно знания за промяна на информацията за **метриката** за маршрутизиране, нарушителят е в състояние да промени пътя по такъв начин, че неговият достъп да стане видимо по-добър през оставената „задна вратичка“. Тази модификация може да доведе до пренасочването на целия трафик от мрежата на нарушителя, задръствайки връзката през задната вратичка, и предизвиквайки отхвърлянето на повечето трафик. Това е един краен и преднамерен пример за въплъщаване.

Отказът на услуги (DoS) е прекратяване на услугите или защото системата е унищожена, или защото временно не е на разположение. Примерите включват унищожаването на твърдия диск на компютър, прекъсване на физическата инфраструктура и заемането на цялата налична памет.

Много обичайни DoS атаки са подбудени от мрежови протоколи, например IP.

DDoS

През последните години, един вариант на DoS атаките причини дори повече проблеми. Това е атаката *на разпределения отказ на услуги (DDoS)*, при която се използват множество машини за започване на DoS атака. Основите на DDoS атаката са показани на фигура 2.5.



Фиг. 2.5 Основни положения при DDoS атака

DDoS клиентът се използва от човек, дирижиращ атака, и служи за първоначална точка. Посредникът представлява компрометиран хост, на който работи специална програма. Всеки посредник е способен да контролира множество агенти. *Агентът* представлява компрометиран хост, на който също

работи специална програма. Всеки агент е отговорен за генерирането на поток от пакети, насочени към желаната жертва.

Много от тези атаки вече са полуавтоматични или изцяло автоматични. При полуавтоматичните DDoS атаки, нарушителят обикновено използва автоматични инструменти за сканиране и компрометиране на уязвими машини и за заразяване на тези машини с кода за атака. В даден по-късен момент, машините с кода за атака се използват за стартирането на широко-разпростряна атака. Още по-проблемни са напълно автоматичните атаки, при които нуждата за последваща комуникация с атакуващите машини се избягва. Кодът за атака, използван за заразяване на машините, вече съдържа часа, в който ще започне атаката, типа на атаката, както и предварително програмирана продължителност и цели.

2.3 Видове атаки

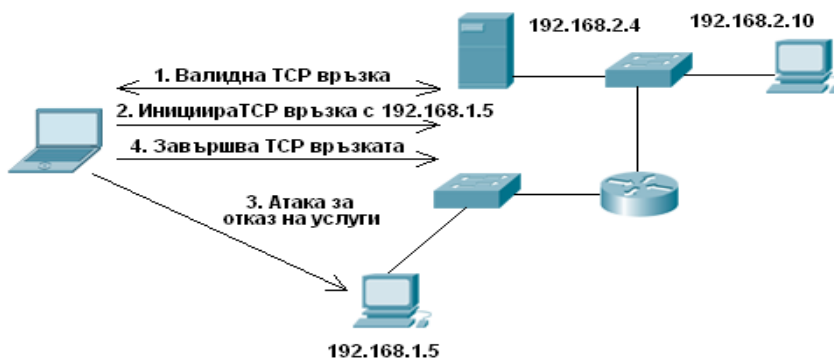
Атаките се възползват от слабости в системите. Тези слабости могат да бъдат причинени от лошо проектирани мрежи или от лошо планиране. Добро правило е да се попречи на всякакви неоторизирани системи да получат достъп до мрежата, където могат да се използват слабости в продукти и технологии.

Атаките с *измама* са добре известни в Интернет света. *Измамването* включва предоставянето на фалшива информация за идентичността на дадена личност или хост, за да се получи неоторизиран достъп до някоя система. Измамването може да се постигне дори и само чрез генерирането на пакети с лъжливи адреси на източниците или чрез възползването от известно поведение на някоя слабост на даден протокол.

Тъй като пълното разбиране на комплекта IP протоколи е ключов елемент при повечето атаки, затова ще бъде описан комплекта протоколи заедно със слабостите на всеки от тях (например TCP, ICMP, UDP, DNS, NNTP, HTTP, SMTP, FTP, NFS/NIS и X Windows).

Атака с TCP/IP поредни номера

Когато някой атакуващ знае шаблона за поредните номера, е доста лесно да се представи за друг хост. Фигура 2.6 показва подобен сценарий.



Фиг. 2.6 Подправяне на TCP/IP поредни номера

Тъй като поредните номера не се избират случайно (и не нарастват случайно), тази атака проработва - въпреки че са необходими известни умения, за да се извърши.

Съществува поправка за TCP в RFC 1948, включваща разделянето на областта за поредните номера. Всяка връзка притежава собствена област с поредни номера. Поредните номера все още нарастват, както преди, но вече няма очевидна или подразбираща се връзка между номерирането в тези области.

Най-добрата защита срещу фалшифициране е да се включат филтри на пакети на входните и изходните точки на мрежите. Филтрите по външните входни точки трябва изрично да отхвърлят всякакви входящи пакети (пакети, идващи от външния Интернет), които твърдят, че са изпратени от хост от вътрешната мрежа. Филтрите по вътрешните изходни точки трябва да разрешават само изходящи пакети (пакети, насочени от вътрешната мрежа към Интернет), които са издадени от хост във вътрешната мрежа.

Отвличане на TCP/IP сесия

Отвличането на сесия е специален случай на TCP/IP измама и е много по-лесно от фалшифицирането на поредни номера. Нарушителят следи сесия между два комуникиращи хоста и инжектира трафик, който изглежда все едно идва от един от тези хостове, открадвайки на практика сесията от един от хостовете. Законният хост е отхвърлен от връзката и нарушителят продължава сесията със същите права за достъп, както законния хост.

Отвличането на сесия е много трудно за засичане. Най-добрата защита е да се използват услуги за поверителност и да се криптират данните, за да се подсигурят сесиите.

TCP SYN атака

Когато започне нормална TCP връзка, получаващият хост получава SYN (синхронизация/начало) пакет от хоста-източник и изпраща обратно SYN/ACK (синхронизация потвърдена) пакет. Получаващият хост след това трябва да чуе ACK (потвърждение) за SYN/ACK преди да бъде установена връзката. Тази обмяна представлява трифазовото TCP ръкостискане.

Докато чака ACK за SYN/ACK, опашка на връзката с краен размер при хоста-дестинация следи връзките, чакащи да бъдат завършени. Тази опашка обикновено се изпразва бързо, защото ACK се очаква да пристигне няколко милисекунди след изпращането на SYN/ACK.

Атаката TCP SYN се възползва от това като атакуващият хост-източник генерира към хоста цел TCP SYN пакети със случайни адреси на източник. Хостът-дестинация изпраща SYN/ACK обратно до случайния адрес на източника и добавя нов запис в опашката с връзки. Тъй като SYN/ACK е предназначен за неправилен или несъществуващ хост, последната част от трифазовото ръкостискане никога не завършва и записът остава в опашката с връзки докато изтече зададен таймер - обикновено около минута. Чрез генериране на фалшиви TCP SYN пакети от случайни IP адреси с голяма скорост, нарушителят може да запълни опашката с връзки и да откаже TCP

услуги (например email, пренос на файлове или WWW услуга) на правомерните потребители.

Протоколът UDP

Подобно на TCP, *User Datagram Protocol* (UDP) е протокол от транспортния слой. UDP обаче предоставя ненадеждна услуга без връзка за доставка за пренос на съобщения между машини. Той не предлага поправка на грешки, повторно предаване или защита срещу изгубени или дублирани пакети. UDP е проектиран за простота и скорост, за да се избегнат скъпите разходи, свързани с установяването и поддръжката на връзката.

Тъй като няма контрол над това, колко бързо се изпращат UDP съобщения и няма ръкостискане за установяване на връзка или поредни номера, UDP пакетите са много по-лесни за фалшификация, отколкото TCP пакетите. Поради това е добре да се сложат пакетни филтри на входните и изходните точки на фирмената мрежа, за да се разрешат и забранят изрично UDP-базирани приложения.

Протоколът ICMP

Internet Control Message Protocol (ICMP) се използва от IP слоя за обмен на контролни съобщения. ICMP се използва и от някои популярни диагностични инструменти, като ping и traceroute.

ICMP съобщението се капсулира в IP пакет.

Както е определено от RFC 791, IP пакетите могат да са с максимална дължина от 65 535 (2¹⁶ - 1) байта; тази дължина на пакета включва дължината на хедъра (обикновено 20 байта, ако не са зададени никакви IP опции). Пакети, които са по-големи от максималната транспортна единица (MTU), се фрагментират от предавателя на по-малки пакети, които в последствие се сглобяват отново от сървъра.

MTU е различна при различните видове среди (Таблица 2.3).

Ping of Death

Ping of Death е атака, възползваща се от слабостта за фрагментиране на големи пакети с ICMP echo заявки (т.е. „*ping*“).

Пакетът с ICMP echo заявка се състои от осем байта с информация на ICMP хедъра, следвани от редица байтове с данни на ping заявката. Поради това, максималният позволен размер на областта за данни се изчислява по следния начин:

Проблемът е, че е възможно да се изпрати нелегитимен ICMP echo пакет с повече от 65 507 байта данни, дължащо се на начина на фрагментиране. Фрагментирането разчита на стойност за отместване във всеки фрагмент, за да се определи къде е мястото му при обратното сглобяване.

Временна поправка за защита от Ping of Death е да се блокират *ping* пакети на входните и изходните точки на фирмената мрежа. Идеалното решение е да се защити TCP/IP имплементацията срещу препълване при сглобяването на IP фрагменти.

Спам атака

Голям процент от email атаките са базирани на email бомбардирането или спамването. Email *бомбардирането* се характеризира с нарушители, които многократно изпращат едно и също email съобщение на определен адрес. Email *спамването* е вариант на бомбардирането; то се отнася за изпращането на email до стотици или хиляди потребители (или до списъци, простиращи се до толкова много потребители). Email спамването може значително да утежни ситуацията, ако получателите отговорят на email, предизвиквайки отговор до всички първоначални адреси.

Когато голямо количество email съобщения бъде насочено през един сайт, той може да изпадне в отказ на услуги поради загуба на мрежова свързаност, блокиране на системата или отказ от работа поради някоя от следните причини:

- Претоварване на мрежовите връзки
- Използване на всички налични системни ресурси
- Запълване на диска в резултат на много поща и съответните записи в системните дневници.

Един небезизвестен червей за масово разпращане на поща, наречен W32.Sobig.F, зарази хиляди машини през август 2003 г. Червеят използваше своя собствена SMTP машина, за да се размножава и саморазпраща до множество адреси. Фалшивите адреси From и Send To се взимаха от файловете, намерени на компрометираните компютри.

Сигурност на безжични мрежи

Безжичните мрежи са се превърнали в една от най-интересните цели за пробиви в сигурността. Повечето устройства за безжични мрежи се доставят с изключени възможности за защита, а няколко уеб сайта документират всички национални безплатни безжични връзки, давайки шанс на потенциалните натрапници избор. Въпреки че много нарушители просто експлоатират тези „безплатни“ връзки като средство за безплатен Интернет достъп или за скриване на своята идентичност, останалите може да сметнат тази ситуации като възможност да пробият мрежи, които в противен случай биха били трудни за атакуване от Интернет, защото за разлика от жичните мрежи, безжичните изпращат данни по въздуха и често излизат извън физическите граници на организацията.

Отказ на услуги при безжични мрежи

Лесно е да се попречи на безжичните комуникации. Един елементарен предавател може с лекота да предизвика DoS атака и да направи комуникацията невъзможна. Изпращайки например постоянно заявки за достъп до дадена AP, независимо дали са успешни или не, накрая ще доведе до изчерпване на наличния радио-честотен спектър и ще я направи недостъпна. Други безжични услуги в същия честотен обхват също могат да намалят спектъра и използваемата честотна лента на технологията за безжичен LAN.

Въпреки че въобще нямат намерение да предизвикват отказ на услуги, хората, които внедряват безжични мрежи, трябва да са наясно и за други пречки, които евентуално биха направили мрежата неизползваема, ако бъдат

добавени към офис средата. В това число влизат микровълнови печки близо до кухни, метални шкафове и евентуално голямо индустриално оборудване.

WEP несигурност

Стандартите 802.11 дефинират WEP като прост механизъм за защита на комуникацията между безжични LAN AP и мрежови интерфейсни карти (NIC). WEP използва алгоритъма за криптиране RC-4 и изисква един и същ ключ да се споделя от всички комуникиращи страни. За да се избегне с ограниченията за износ от САЩ, които важаха при разработката на стандарта, IEEE 802.11b изисква 40-битови ключове за криптиране, въпреки че много производители вече поддържат незадължителния 128-битов стандарт. Тези дължини на ключове не са много силни и WEP може лесно да бъде пробит, както в 40-битовия вариант, така и в 128-битовия посредством готови инструменти, свободно достъпни в Интернет.

Обобщение

Настоящата глава разгледа различните заплахи за корпоративната мрежа чрез детайлното представяне на обичайните видове атаки и слабости, както и мерките, които могат да бъдат предприемат на ниво политика, за да се гарантира някаква степен на сигурна мрежа.

Видовете заплахи обикновено са под формата на неоторизиран достъп, въплъщаване или DoS. Ако разберете някои от подбудите за дадена атака, може да разберете кои части от мрежата са уязвими и какви действия може да предприеме нарушителя.

Най-обичайните слабости са представени в подробности, за да се помогнат администраторите да оценят податливостта на мрежата. Това може да се окаже безценно при определянето на стъпките, които трябва да бъдат предприети, за да се предпазят най-изложените области на мрежата.

Глава трета

Принципи на контрол на трафика

Проектирането и разработването на една мрежа е свързано с множество фактори, които трябва да бъдат отчетени.

На първо място е необходимо да се реши каква връзка с външния свят (Интернет) ще бъде реализирана. Този елемент от проектирането се обвързва с така наречената гранична мрежа (връзката между вътрешната (локалната) и външната (Интернет) мрежа). Подбирането на правилното решение за такава гранична мрежа може да попречи на цели групи от атаки да достигнат защитените сървъри. Трябва да се отбележи, че граничната мрежа може да предотврати използването на пробита система от вътрешната мрежа за атакуване на други системи. Следователно проектирането на сигурни мрежи е ключов елемент в управлението и удържането на риска.

На второ място е необходимо да се разработи правилната вътрешна инфраструктура на локалната мрежа с цел да се намали елемента на атака и подsigуряване на мрежата от недоброжелатели, служители на фирмата.

Най-често срещаната в днешни дни архитектура на защитна стена е показана на фигура 3.1. Архитектурата включва маршрутизатор, който филтрира пакетите и действа като наша първоначална, но не и единствена, защитна линия. Директно зад този маршрутизатор е разположена "същинската" защитна стена, изградена с правила на iptables.

Към вътрешната мрежа не съществува директна връзка от Internet.



Фиг. 3.1 Проста архитектура на защитна стена

Всички външни маршрутизатори трябва да използват някакъв вид филтриране на пакети, т. нар. "списъци за контролиране на достъпа" (Access Control List, ACL). Даже и когато при следващата стъпка навътре от такъв маршрутизатор се достига до защитна стена, наличието на допълнителна защита никога не е вредно.

Най-малкото, което маршрутизаторите, свързани към Интернет, трябва да направят е да отхвърлят пакети, чийто изходен или краен IP-адрес не може да се

открие в Интернет, понеже такива пакети със сигурност могат да се приемат за "фалшифицирани" (подправени).

Тази архитектура е често използвана, но не е най-добрата.

На фигура 3.2 е показана оптимална архитектура на защитна стена/ДМЗ.



Фиг.3.2 Архитектура на ДМЗ с една защитна стена

В случая за защитна стена се използва машина с три мрежови интерфейса. Машините, които осигуряват публично достъпни услуги, образуват отделна мрежа със собствена връзка към защитната стена, а останалата част от мрежата на организацията използва друг интерфейс на защитната стена. Ако защитната стена се конфигурира по подходящ начин, тя ще може да използва различни правила за оценяване на трафика:

- От Интернет към ДМЗ
- От ДМЗ към Интернет
- От Интернет към вътрешната мрежа
- От вътрешната мрежа към Интернет
- От ДМЗ към вътрешната мрежа
- От вътрешната мрежа към ДМЗ

На пръв поглед изглежда, че всичко това ще изисква далеч повече администраторска работа отколкото алтернативата с разполагането на услугите върху защитната стена или във вътрешната мрежа, но тази архитектура всъщност е потенциално много по-проста, понеже ДМЗ може да се третира като един-единствен логически елемент. В случая с поставянето на услугите във вътрешната мрежа всяка машина трябва да бъде разглеждана отделно от останалите (освен ако всички услуги не са разположени на една мрежа, чийто адрес е неотличим от другите части на мрежата).

Разпределяне на ресурсите в ДМЗ

Машините в ДМЗ могат да бъдат атакувани както от страна на Интернет, така и от страна на машините във вътрешната мрежа. Следователно трябва да се

помисли не само как да се предотврати пробиването на някоя система в ДМЗ, но също така какви биха били последствията в такъв случай.

Едно от тези потенциални последствия е използването на пробитата машина за подслушване на трафика в ДМЗ.

Както вече беше казано мрежите от тип ДМЗ са по добри, защото те помагат публично достъпните системи да бъдат изолирани, но това не означава, че тези системи трябва да бъдат по-податливи на атака.

Комутаторите също така предлагат по-добра производителност от концентраторите: през повечето време всеки порт си има свой капацитет на връзката, вместо да споделя общата връзка с всички останали портове. Всеки комутатор си има лимит, показващ реалното количество пакети, които може да обработи: един комутатор с 10 порта от по 100 Mbps в действителност не може да работи със скорост 1000 Mbps, ако лимитът му е 800 Mbps. Въпреки това дори и слабите комутатори многократно надвишават по производителност съответстващите им концентратори.

Отговорите на другите два въпроса за разпределянето на услугите в една ДМЗ обикновено могат да се определят от фактори, независещи от сигурността (цена, очаквано натоварване, ефективност и т.н.), при положение, че всички машини в ДМЗ са надлежно подsigурени и наблюдавани и че правилата на защитната стена (филтриране на пакети, настройките на посредническите услуги и т.н.), управляващи трафика към и от ДМЗ, са възможно най-строги.

Защитната стена

Естествено е, че за да се изгради добра гранична мрежа, ще трябва да бъде направено нещо повече от това да се създаде и запълни една ДМЗ.

Това, което в крайна сметка разграничава ДМЗ от вътрешната мрежа, е ***защитната стена***.

Защитната стена (или защитни стени) има първата и последната дума по въпроса дали дадени данни могат да влязат или напуснат всяка от вашите мрежи. Въпреки че ще е погрешно да се счита изградената защитна стена за универсалното решение на всички проблеми със сигурността, е важно защитните стени да бъдат внимателно настроени, съвместно поддържани и под постоянно наблюдение.

Просто филтриране на пакети

Защитните стени с обикновено филтриране на пакети оценяват пакети, основавайки се единствено на техните заглавни части (фигура 3.4). Това е един относително бърз начин за регулиране на трафика, но той също така е и лесен за заобикаляне. Атаките, в които се фалшифицира IP-адресът на подателя, обикновено не се блокират от филтрирането на пакети и понеже одобрените пакети преминават директно през защитната стена, пакетите с "легитимни" заглавни части, носещи опасни данни (като атаки за препълване на буфера), често могат да бъдат изпратени непроменени към "защитени" цели.

Филтриране на пакети с отчитане на състояние

Съществуват два вида филтри на пакети с отчитане на състояние: обикновен и Check Point.

Обикновен тип

При този тип филтриране се проследяват връзките по TCP, започвайки от "трипосочното ръкостискане" (SYN, SYN/ACK, ACK), което се осъществява в самото начало на връзката, и завършвайки с последния пакет от сесията (FIN или RST).

Проверяват се: IP-адресът и портът на подателя, IP-адресът и портът на получателя и поредните номера по TCP на всеки пакет.

Този механизъм има няколко важни предимства пред просто филтриране на пакети без отчитане на състояние. Първото от тях е двупосочността: ако филтрирането на пакетите не отчита по някакъв начин състоянието на връзката, защитната стена няма как да знае дали даден входящ пакет е част от вече съществуваща връзка (например създадена от вътрешна машина) или е първият пакет от нова (входяща) връзка. На обикновените филтри на пакети може да бъде указано да приемат, че всеки пакет по TCP с вдигнат флаг ACK е част от вече установена сесия, но това отваря врата за многобройни атаки с фалшифициране.

Stateful Inspection

Технологията на Check Point представлява хибрид между филтрирането на пакети и посредничеството на приложно ниво. Но поради различията в начина на третиране на отделните услуги, истинската сила на Stateful Inspection вероятно е много по-близка до тази на обикновените филтри на пакети, отколкото до най-добрите посреднически защитни стени (т.е. защитни стени с приложни шлюзове).

Въпреки че Stateful Inspection е запазена марка на Check Point, други защитни стени (като PIX на Cisco и дори iptables на Linux) също предлагат подобна способност за вземане на решения на приложно ниво при проследяването на сесиите на определени видове приложения.

Посредници на приложно ниво

Третият тип обичайни технологии за защитни стени е посредничеството на приложно ниво. За разлика от филтрите със и без отчитане на състояние, които преглеждат, но не променят пакетите (освен че в някои случаи ги пренасочват или променят адресите им), една защитна стена с посредничество на приложно ниво участва активно като междинна стъпка във всички транзакции, минаващи през нея (фиг. 3.6).



Фиг. 3.6. Посредник на приложно ниво

Този тип защитни стени често са наричани "посредници на приложно ниво", защото за разлика от другите видове междинни сървъри, които увеличават производителността, но не винаги и сигурността, защитните стени обикновено разполагат с голямо количество специфична информация за услугите, които минават през тях.

В областта на защитните стени не трябва да се бъркат понятията посредниците на приложно ниво ("приложни шлюзове") с "препредаващите посредници".

Първите притежават специфична за приложенията информация, а вторите не. "Преподаващите посредници" (като продуктите, базирани на SOCKS) възпроизвеждат данните от подателя към получателя, но не ги обработват или регулират, както правят приложните шлюзове.

Избор на защитна стена

Изборът на вида защитна стена, хардуерната платформа, на която ще бъде разположена, и на самия комерсиален или безплатен пакет за защитна стена, който ще използвате, зависи от нуждите на организацията, финансовите и техническите ресурси и донякъде от чисто субективни предпочитания.

Например организация, която трябва да пази целостта на своите данни колкото е възможно по-строго (защото на риск ще са изложени съответно информацията за клиентите), най-вероятно ще се нуждае от защитна стена с посредничество на приложно ниво. Ако непрекъснатата поддръжка е от значение, един комерсиален продукт ще бъде най-добрият избор.

От друга страна едно училище може да не разполага с техническите ресурси (т.е. професионални мрежови специалисти), нужни за поддръжане на посредническа защитна стена. Също така е твърде вероятно училището да няма финансовата възможност да закупи и поддържа комерсиален продукт от високо ниво. За такава организация по-доброто решение ще бъде една не особено скъпа защитна стена с филтриране на пакети (с отчитане на състояние) или дори защитна стена на Linux или FreeBSD

В общия случай приложните шлюзове предлагат най-добра защита, но те са по-сложни за администриране и имат по-големи изисквания към скоростта и капацитета на хардуера. Защитните стени с филтриране на пакети с отчитане на състояние придвижват пакетите по-бързо и са по-прости за администриране, но обикновено предлагат много по-добра защита на някои услуги, отколкото на други. Обикновените филтри на пакети са най-бързи и най-евтини, но също така и най-лесни за заобикаляне.

Подсигуряване операционната среда на защитната стена

Първо, преди да се инсталира софтуера на защитната стена, би трябвало да подсигури операционната среда, върху която ще бъде тя разположена. Ненужният софтуер трябва да бъде премахнат; ненужните скриптове трябва да бъдат изключени; важните демони трябва да се стартират без администраторски привилегии и, ако е възможно, с променена основна директория (чрез chroot); софтуерът на ОС и всички приложения трябва да следват текущите версии и кърпки. Колкото е възможно по-скоро след инсталиране на ОС (и преди

системата да бъде свързана към Интернет), трябва да се инсталира и инициализира програма за проверка на цялостта като *tripwire* или *AIDE*.

Освен това ще трябва да бъде решено кой ще има административен достъп до защитната стена и особено кой ще може да променя или създава политиката на защитната стена. Никой администратор не бива да получава привилегии с по-високи права за достъп, ако наистина не се нуждае от тях.

Например администраторът, който периодично прави резервни копия на системата, трябва да има акаунт, който да му дава достъп за четене до всички файлови системи, подлежащи на архивиране, но не и достъп за писане. Освен това неговият акаунт не трябва да принадлежи към групите wheel или root (т.е. той не трябва да може да изпълнява su root).

Конфигуриране на правила против подправяне на IP

Защитната стена трябва да предлага механизми срещу подправяне на IP-адреси, които трябва да се настроят.

Много мрежови атаки включват фалшифицирани пакети, т.е. пакети с подправен IP-адрес на подателя. Тази техника се използва най-вече в атаките за отказване от услуга (Denial of Service - DoS), за да се прикрие източникът на атаката, както и като опит пакетите да се представят за произхождащи от доверени (вътрешни) мрежи. Способността за засичане на фалшифицирани пакети е толкова важна, че задължително защитната стена трябва да има такава функционалност.

Командите на iptables за блокиране на подправени IP-адреси ще бъдат следните:

```
iptables -I INPUT 1 -i eth0 -s 192.168.0.0/16 -j DROP
```

```
iptables -I INPUT 2 -i eth0 -s 10.0.0.0/8 -j DROP
```

```
iptables -I INPUT 3 -i eth1 -s ! 192.168.100.0/24 -j DROP
```

```
iptables -I INPUT 4 -i eth2 -s ! 10.0.0.0/8 -j DROP
```

Повечето защитни стени, включително и iptables на Linux 2.4, могат да бъдат настроени да отказват пакети по два различни начина.

Първият метод (обикновено наричан отказване, Dropping) е пакетите да се откажат "тайно", т.е. без да се уведомява техният подател.

Вторият метод (обикновено наричан отхвърляне, Rejecting) е на подателя да се върне пакет на TCP с вдигнат флаг RST, ако отхвърлената заявка е била направена по протокола TCP, или съобщението "Port Unreachable" на ICMP, ако заявката е била получена по UDP.

Стриктно ограничаване на трафика, излизащ от ДМЗ

Едно от основните предположения за ДМЗ е, че машините в нея са изложени на значителен риск от пробиване.

За да се намали този риск, трябва да се ограничи излизания от ДМЗ трафик да използва единствено необходимите и стандартните услуги и портове.

Така например един веб-сървър в ДМЗ трябва да може да приема сесии по HTTP на порт 80, но не е нужно той да може и да създава сесии по TCP 80 и не трябва да му се позволява да прави това. Ако този сървър бъде заразен по някакъв начин с вируса Code Red (примерно), всички опити на Code Red да

използва вашия сървър за идентифициране и заразяване на други системи ще бъдат блокирани.

Внимателно трябва да се обмисли какъв трябва да бъде трафикът от ДМЗ към вътрешната мрежа. Това предполага правилно проектиране на работната среда така, че да се намалят до минимум нуждата от такъв трафик.

Например, ако дадена машина в ДМЗ трябва да може да отправя заявки по DNS, тя трябва да се настрои да използва DNS-сървъра в ДМЗ (ако има такъв), вместо вътрешния DNS-сървър.

Един сървър в ДМЗ със зле контролиран достъп до вътрешната ви мрежа представлява огромна заплаха за сигурността на цялата мрежа.

Ограничаване на вътрешните системи за достъп навън

Има две причини, поради които трябва да се ограничи достъпа на вътрешната мрежа до външни услуги.

Първо, ограниченията помагат да се запази скоростта на връзката към Интернет. Разбира се, понякога ще бъде възможно потребителите да заобикалят ограниченията на защитната стена, като теглят аудио потоците през TCP 80, но последиците в този случай ще бъдат различни от тези, когато достъпът навън не се ограничава.

Второ (също като при системите в ДМЗ), ограничаването на достъпа, който вътрешните системи имат навън, помага за намаляване на риска една пробита вътрешна система да бъде използвана за атака на машини от други мрежи (особено в случаите, когато нападателят е вирус или друг вид злонамерен код).

Използване на приложен шлюз

Ако организацията разполага с техническите средства и може да отдели достатъчно хардуерни ресурси, защитните стени от тип "приложен шлюз" предлагат по-добра защита от най-добрия филтър на пакети с отчитане на състояние.

Посредникът на приложно ниво ще може да се използва само за някои услуги и филтриране на пакети за останалите.

Посредническите защитни стени почти винаги позволяват да се използва известно количество филтриране при необходимост.

Сигурност

Относно защитните стени никога не трябва да си мислите, че защитната стена предлага абсолютна сигурност. Единствената абсолютна защита от мрежови атаки е срязаният мрежови кабел.

Защитната стена трябва да бъде конфигурирайте колкото е възможно по-внимателно.

Публично достъпните сървъри в ДМЗ трябва да се подсиgurят така, че все едно изобщо нямате защитна стена, т.е. ако нещо се случи със защитната стена те да имат достатъчно висока сигурност.

Глава четвърта Мрежово администриране

4.1. Принципи на мрежовото администриране

Компютърната операционна система е софтуерът, който осигурява основата, върху която работят приложенията и услугите. По подобен начин една *мрежова операционна система (network operating system — NOS)* позволява на устройствата да комуникират помежду си, както и да споделят ресурси по мрежата.

Мрежовата операционна система понякога се използва за описание, на която и да е операционна система с вградени мрежови компоненти. Това е противоположно на *самостоятелната* операционна система, проектирана да се използва в изолация. Все пак мрежовата операционна система си е операционна система, като NetWare или NT Server, работеща на мрежов сървър.

Една сървърно-базирана мрежа притежава един или няколко посветени сървъра. Сървърът обикновено е (но не задължително) машина с по-бърз процесор, голямо количество RAM и по-голям твърд диск или дискове, отколкото работните станции, които се свързват с него.

Клиент-сървър средата се различава от равноправната (peer-to-peer) мрежа.

Мрежата от тип клиент-сървър предлага среда, която е лесна за администриране, архивиране и обезопасяване. В такава мрежа клиентите заявяват данни и ресурси от сървъра. В този случай сървърът може да споделя само файлове, но е възможно да споделя и принтери, приложения, модеми, Интернет връзка и други ресурси.



Фигура 4.1 Средите работна група, клиент/сървър и мейнфрейм.

Този принцип на обработка с клиент/сервър приложение е различен в сравнение с този, в който потребителите създават бази данни в Access (работещ на техните локални машини), след което съхраняват базите данни на файлов сервър. Когато се направи заявка, цялата база данни се сваля на клиентската машина и процесът на търсене се провежда там. Става ясно, че пропускателната способност на мрежата се пести при метода с клиент/сервър приложение.

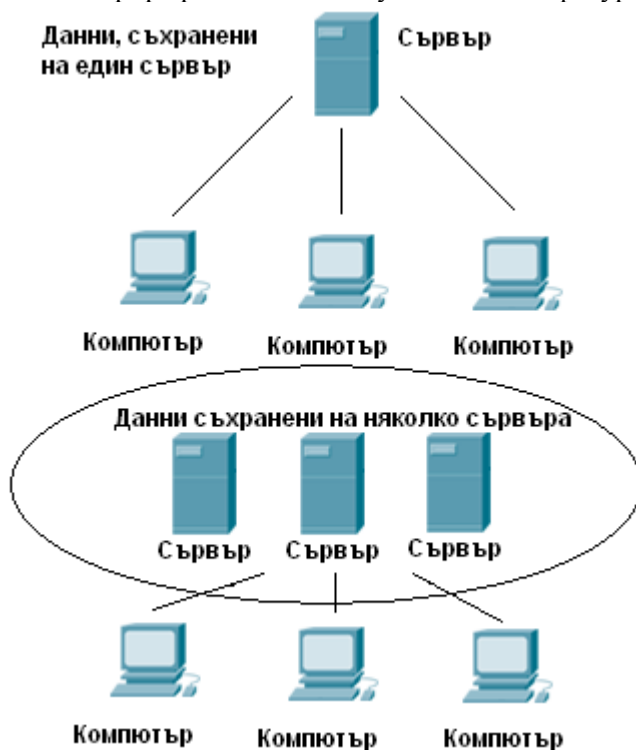
Недостатъкът на клиент/сервър приложенията е първоначалната цена. Програмите за клиент/сервър бази данни, като SQL Server или Oracle, са относително скъпи и могат да не са ценово-ефективни за малки мрежи, където заявките към базите данни са прости и не много чести.

Конфигурация на мрежа клиент/сервър

Клиент/сервър мрежата може да бъде конфигурирана по един от следните два начина:

- Данните могат да бъдат разположени на един сервър за бази данни.
- Данните могат да бъдат разпределени (distributed), или разпръснати, по множество сервъри за бази данни.

Фигура 4.2 илюстрира разликата между тези две конфигурации.



Фиг. 4.2 Данните могат да са разположение на един сервър или да бъдат разпределен на множество сервъри.

Работна станция

Едно *хранилище за данни (data warehouse)* представлява централно място, където се съхраняват огромни количества данни. По традиция хранилищата за

данни бяха базирани на мейнфреймове, но някои мрежови операционни системи за РС-та са проектирани специално за тази цел. В хранилището данните могат да бъдат съхранявани на масив от дискове, инсталирани на един сървър, или да бъдат разпръснати на дисковете на множество сървъри във *ферма от сървъри (server farm)*.

Споделяне на мрежови ресурси









Процесът на споделяне по мрежата зависи от мрежовата операционна система. Някои мрежови операционни системи, като тези от фамилията Windows, по подразбиране не споделят абсолютно нищо. Ако е необходимо да дадени ресурси да има достъп по мрежата, те трябва изрично да се споделят в процес, наречен *създаване на споделен ресурс (creating a share)*.

Други мрежови операционни системи, като NetWare, се държат точно по обратния начин - ресурсите са споделени по подразбиране.

За да може един Windows компютър да споделя своите ресурси, услугата **File and Print Sharing** трябва да е разрешена. Ако компютърът използва Windows 2003, там трябва да е инсталирана и да работи *услугата server*.

Сигурността на ниво споделен ресурс (share-level security) по принцип се използва в равноправните (peer-to-peer) мрежи, където машините обикновено работят под операционна система като Windows. При този тип сигурност, когато се избере за споделяне някой ресурс по мрежата, например папка, се задава парола за този ресурс. За да може някой да осъществи достъп до споделената папка, той трябва да знае и да въведе вярната парола, когато бъде подканен да направи това.

Всеки споделен ресурс има различна парола, като тези пароли трябва да бъдат дадени на всички хора, които са авторизирани да осъществяват достъп до съответните споделени ресурси (фиг.4.3)

Споделени папки	Парола	Потребителят трябва да помни пароли
За достъп от Ivan, Maria  Папка 1	1q2w3e	 Ivan Папка 1 - 1q2w3e Папка 2 - 3q4w5e
За достъп от Ivan, Ivo  Папка 2	3q4w5e	 Maria Папка 1 - 1q2w3e Папка 3 - 5f6g7y Папка 4 - S4d2f5
За достъп от Ivo, Maria  Папка 3	5f6g7y	 Ivo Папка 2 - 3q4w5e Папка 3 - 5f6g7y
За достъп от Ana, Maria  Папка 4	S4d2f5	 Ana Папка 4 - S4d2f5

Фиг. 4.3 Споделени ресурси с пароли

Сигурността на ниво потребител (user-level security) е много по-лесна за управление в средна или голяма мрежа, отколкото сигурността на ниво споделен ресурс. При сигурността на ниво потребител всеки потребител има *потребителски акаунт (user account)*, който е защитен с парола. Потребителят се логва (влиза в) компютъра с този акаунт. Всеки споделен ресурс се конфигурира така, че достъпът до него да е разрешен само за авторизираните потребители. Когато даден потребител се опита да осъществи достъп до ресурс, се проверява *списъкът за контрол на достъпа (access control list)*, асоцииран с ресурса. Този списък съдържа авторизираните акаунти, които имат позволения за съответния ресурс. Ако акаунтът, с който е влязъл потребителят, е в списъка, тогава потребителят може да осъществи достъп.

Управление на мрежови акаунти

Сигурността на ниво потребител изисква създаване на потребителски акаунт.

Повечето мрежови операционни системи позволяват организиранети и поддръждането на тези потребителски акаунти в *групи* за по-лесното им управление. Някои мрежови операционни системи също така изискват всеки компютър, който се логва в мрежата, да има *компютърен акаунт* (наричан също *машинен акаунт*). Това осигурява допълнително ниво на защита и подобрява сигурността на мрежата.

Ще бъдат разгледани трите типа акаунти: потребителски, групови и компютърни.

Потребителски и групови акаунти в Windows XP

Windows XP осигурява потребителски акаунти и групови акаунти, на които потребителите могат да бъдат членове.

Потребителските акаунти са предназначени за отделни потребители. Предназначението на груповите акаунти, обикновено наричани *групи*, е да опростяват администрирането на множество потребители.

В системата може да се влиза с потребителски акаунти, но не и с групови акаунти.

Локални потребителски акаунти

• Локални акаунти

Потребителските акаунти, дефинирани на локалния компютър, се наричат *локални потребителски акаунти*. Тези акаунти имат достъп само до локалния компютър и трябва да предоставят потребителско име и парола, за да получат достъп до мрежовите ресурси. Създаването на локални потребителски акаунти се извършва чрез помощната програма Local Users And Groups.

• Домейн-акаунти

Потребителските акаунти, дефинирани в услугата Active Directory, се наричат *домейн-акаунти*. Чрез възможността Single Sign-On (единствено влизане) тези акаунти имат достъп до ресурсите в целия домейн. Домейн-акаунтите се създават в помощната програма Active Directory Users and Computers.

Всички потребителски акаунти се идентифицират с logon-име (името, с което потребителят влиза в системата).

Това име се състои от две части:

- **Потребителско име** Текстово означение на акаунта
- **Работна група или домейн на потребителя** Работната група или домейнът, където съществува потребителят

Когато се инсталира Windows, операционната система инсталира подразбиращи се потребителски акаунти.

Основните акаунти, които се срещат, са следните:

- **Administrator** Administrator е предварително дефиниран акаунт, който осигурява пълен достъп до файловете, директории, услугите и другите средства на операционната система. Този акаунт не може да бъде изтрит или забраняван. В Active Directory акаунтът Administrator има достъп и привилегии за целия домейн. В локалната работна станция акаунтът Administrator обикновено има достъп само до локалната система.

- **Guest** Guest е предназначен за потребители, които се нуждаят от достъп еднократно или от време на време. Въпреки че този акаунт има ограничени системни привилегии, администратора в мрежата трябва да е много внимателен, ако той се използва, понеже системата ще бъде отворена за потенциални проблеми със сигурността. Рискът е толкова голям, че акаунтът е първоначално забранен, когато се инсталира Windows.

- **HelpAssistant** Интерактивната поддръжка на Windows е изградена върху фреймуърк за приложения, който позволява на потребителите да инициират сесии за отдалечено съдействие. Техниците, които отговарят на заявките за отдалечено съдействие, влизат в системата на потребителя чрез акаунта HelpAssistant. Той има ограничени системни възможности, които му позволяват да влиза локално в системите посредством Terminal Services.

- **Support** Акаунтът Support се използва от вградената услуга Help And Support. Този акаунт е член на HelpServicesGroup и има правото Log On As Batch Job (влизане като пакетно задание). Това потребителско право позволява на акаунта Support да изпълнява пакетни актуализации. Форматът на името на акаунта е Support_<id>, където <id> представлява някаква стойност, като например Support_388945a0.

Групови акаунти

Освен потребители, Windows осигурява и групи. Те се използват за предоставяне на права на потребители от сходни типове и за опростяване на администрирането на акаунтите. Членовете на групата имат достъп до ресурсите, които са достъпни за нея. По този начин може да бъде даван на потребителите достъп до различни ресурси, като потребителите се добавят като членове на правилните групи.

В системата може да се влиза с потребителски акаунт, но не и с групов акаунт.

Тъй като е възможно различните домейни или работни групи на Active Directory да имат групи с едно и също име, групите често се указват или като домейн\група, или като работна\група\група

Windows XP използва следните три типа групи:

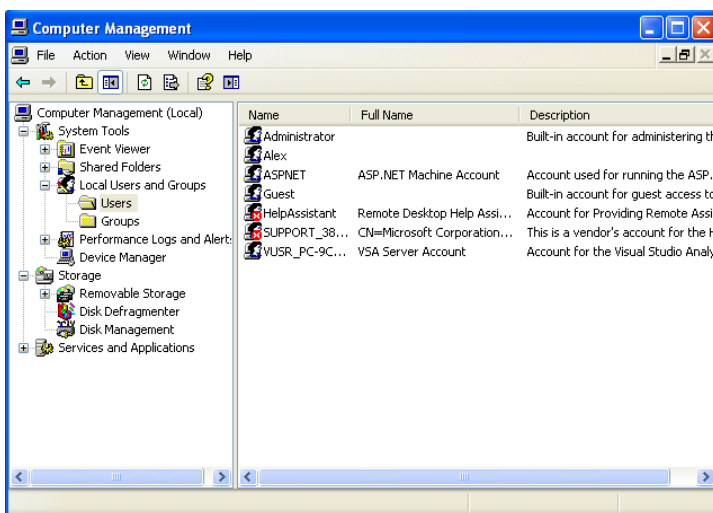
- **Локални групи (local groups)** Дефинират се на локалния компютър и се използват само там. Локалните групи се създават с помощната програма Local Users And Groups.

- **Групи за сигурност (security groups)** Могат да имат асоциирани с тях дескриптори за сигурност. Групите за сигурност се дефинират в домейните посредством Active Directory Users and Computers.

- **Групи за разпространение (distribution groups)** Използват се като списъци за разпространение на електронна поща. Те не могат да имат асоциирани с тях дескриптори за сигурност. Групите за разпространение се дефинират в домейните чрез Active Directory Users and Computers.

Създаване на локални потребителски акаунти

Локалните потребителски акаунти се създават посредством Local Users And Groups. Достъпа може да се реализира през Control Panel. След това се избира Administrative Tools и накрая избор на Computer Management (фиг.4.7).



Фиг.4.7 Създаване на локални потребителски акаунт

Тук могат да се зададат името на потребителя, неговата парола, до кога ще важи тя, дали потребителя ще може да я сменя или не.

Посредством Local Users And Groups могат да бъдат реализирани множество операции:

- Създаване на локални групи за работни станции
- Добавяне и премахване на членове на локални групи
- Разрешаване на локални потребителски акаунти
- Преименуване на локални потребителски и групови акаунти
- Изтриване на локални потребителски и групови акаунти

Изтриването на даден акаунт го премахва за постоянно. След като бъде изтриет даден акаунт, не може да бъде създаван друг акаунт със същото име, който автоматично да получи същите разрешения. Това е така поради факта, че SID идентификаторът на новия акаунт няма да съвпада с този на стария.

Компютърни акаунти

Мрежовите операционни системи, проектирани за среди с висока сигурност, могат да изискват не само потребителите да имат акаунти за логване в мрежата, но и свързаните към мрежата компютри да притежават машинни акаунти. Например в един Windows домейн администраторът трябва да е създал компютърен акаунт за всяка система, работеща под Windows NT или Windows 2003, преди тя да може да се присъедини към домейна.

Операционната система използва компютърния акаунт, за да валидира идентичността на компютъра и да извършва одит на действията, извършвани по време на използване на компютърния акаунт.

Управление на споделени ресурси

В някои мрежови операционни системи, като например тези от фамилията Windows, когато се създава споделен ресурс, той автоматично се споделя по подразбиране с групата Everyone, която съдържа *всички* потребители. Ако това не съответства на това което се желае е необходимо да се модифицират позволенията, когато се създава желан споделен ресурс.

На споделените ресурси обикновено се задават *имена на споделени ресурси (share names)*, които могат да бъдат - но не задължително - същите като действителните имена на самите ресурси. Например, ако се споделя папка, наречена salesdocs, за име на споделения ресурс можете да зададете Sales Documents; това *не* променя името на самата папка.

След като ресурсите бъдат споделени, те трябва да се управляват.

4.2 Мрежови операционни системи

Изборът на мрежова операционна система (NOS) може да бъде сложно решение. Всяка популярна NOS си има силни и слаби страни, и понеже мрежовите операционни системи по принцип са много по-скъпи от настолните (често с цена няколко стотин долара, в зависимост от броя на клиентите, които се свързват към сървъра), решението не се взема с лека ръка.

Популярните мрежи, базирани на мрежови операционни системи, са следните:

- Windows NT и Windows 2000(2003,2008) мрежи
- NetWare мрежи
- UNIX/Linux мрежи

Създаване и управление на потребителски акаунти в Windows NT 4.0

Microsoft осигурява административен инструмент, наречен User Manager for Domains, до който осъществявате достъп от домейн контролера и използвате за създаване, управление и премахване на потребителски акаунти от домейна. Стартирането му се извършва чрез последователност от команди: ***Start, Programs, Administrative Tools, User Manager for Domains.***

Инструментът User Manager for Domains позволява на администратора да създава нови потребителски и групови акаунти; да преименува, модифицира и изтрива акаунти; да назначава пароли; да установява политики за акаунтите; да задава рестрикции за потребителите. Рестрикциите включват указване на това, кога и от коя работна станция те могат да се логват.

Всички групи са т.нар вградени подразбиращи се групи за Windows NT 4.0 домейни.

Двата вградени подразбиращи се потребителски акаунта са Administrator (администратор) и Guest (гост).

Администриране на акаунти в Windows 2000

Административните задачи в Windows 2000 използват обща основа - *Microsoft Management Console (MMC)*, Този инструмент използва *snap-in модули* - такива модули, които съдържат инструментите за специфични административни функции. Потребителите и групите се създават и управляват чрез snap-in модула за MMC, наречен Active Directory Users and Computers. Той може да бъде стартиран, като се избере Start, Programs, Administrative Tools, Active Directory Users and Computers .

Windows 2000, за разлика от Windows NT 4.0, позволява да се поставят обекти, като потребители и ресурси, в контейнерен обект, наречен *организационни единици (organizational units - OUs)*.

Може да бъде извършено делегиране на потребител или на група административна власт над всяка OU.

Това позволява много по-голяма грануларност на контрола, отколкото беше възможно в Windows NT 4.0.

Мапване на устройства в Windows мрежу

Мапването на мрежово устройство в Windows е лесно. Можете да го направите по два начина:

- Чрез използване на Windows Explorer
- Чрез използване на командата **net use**

Мапване на устройство в Windows Explorer

За да се мапне устройство с Windows Explorer, е необходимо да се достигне до папката на отдалечената машина с помощта на Windows Explorer.

След уточняване *името на сървъра, име на споделена папка* се отваря менюто Tools, след което чрез избор на Map Network Drive се реализира асоциирането на споделения ресурс и буква за достъп чрез мапване.

Мапнатото устройство се появява под формата на назначена буква на устройство в левия панел на Explorer, заедно с останалите устройства, CD устройства и дяловете от твърдия диск.

След извършване на указаната процедура може да бъде осъществяван достъп до това устройство от Windows Explorer, My Computer, както и от десктопа, ако се създаде shortcut.

Мапване на устройство с командата net use

Друг начин за мапване на устройство в операционните системи Windows е чрез *път, използващ конвенцията за универсални имена (Universal Naming Convention - UNC)*.

Споделен ресурс може да бъде идентифициран чрез следния синтаксис:

\\име_на_компютър\име_на_споделен_ресурс

За да се мапне мрежово устройство към споделения ресурс, въведете следното на командния ред:

net use *буква_на_устройство*:
\\ *име_на_компютър* \ *име_на_споделен_ресурс*

Споделяне на принтери в Windows мрежи

Споделянето на принтери и свързването към тях в Windows мрежа е просто.

За да се сподели принтер, свързан към локален компютър, след избор на иконата Printers and faxes, достъпна от Control Panel, се кликва с десния бутон на мишката върху името на принтера.

Избира се Sharing, след което чрез уточняване на Shared as се въвежда име на споделен ресурс.

Както при мапването на мрежови устройства, има два начина потребителите да се свържат към споделен мрежов принтер: чрез съветника Add Printer Wizard, или чрез командата net use от командния ред.

Microsoft използва съветници (wizards), представляващи систематична поредица от диалогови прозорци.

Когато съветникът приключи, мрежовият принтер се появява във прозореца Printers and Faxes, след което трябва да може да се печата на него от приложенията компютъра така, все едно че това е локален принтер.

Заданията за печат се спулират (съхраняват се в паметта или на твърдия диск) „изчакайки на линия“ да бъдат отпечатани. Списъкът от чакащи задания за печат се нарича принт спул (print spool).

UNIX и Linux мрежи

UNIX мрежовата операционна система съществува от доста години.

Тъй като сорс кодът е отворен (т.е., достъпен е безплатно за всеки, който иска да го модифицира) и понеже е написан на популярния език за програмиране C, то бизнес организациите, академичните институции и дори отделните личности могат и разработват свои собствени версии.

UNIX операционните системи се използват в работни станции от висок клас, като например машините на Silicon Graphics и Sun. UNIX може да работи като операционна система с команден ред или с графичен потребителски интерфейс (graphical user interface - GUI), като X Window.

Администриране на акаунти в UNIX и Linux

Мрежовата информационна система (Network Information System - NIS), разработена от Sun Microsystems, може да бъде използвана за управление на UNIX сървъри. NIS позволява достъп до мрежовите ресурси с едно логване, което ще рече, че потребителят може да се логне веднъж (с едно име на акаунт и една парола), след което да осъществява достъп до ресурсите на компютрите из цялата мрежа.

Важно е да се отбележи, че в UNIX, за разлика от Windows, командите и имената са *чувствителни към регистъра на буквите (case-sensitive)*; това ще рече, че за операционната система файловото име „MyDocs“ не е същото като „mydocs.“ Всъщност тези два файла могат да съществуват в една и съща директория.

Ако Web сървърът, към който се свързва Web браузър, е UNIX-базиран, трябва да се въведе адреса *точно както е показан*. Ако една Web страница е с име „WebSite.html“, а бъде въведено „Website.html“, ще се изведе съобщение за грешка „file not found“ (файлът не е намерен).

Създаване и управление на потребителски акаунти

За да бъде добавен нов потребителски акаунт в UNIX и Linux, трябва да се използва командата **adduser**.

Само потребители от типа на root или supervisor могат да създават нов акаунт. Синтаксисът е следният:

име_на_потребител: /# adduser

Потребителските акаунти се управляват чрез редактиране на файла /etc/passwd.

Някои UNIX/Linux версии осигуряват *скриптове*, които водят в процеса, а файла може да се редактира и ръчно чрез текстов редактор, (vi, Pico и Emacs са популярни текстови редактори за UNIX/Linux.)

Скриптовете са малки програми. Когато се използва скрипт, за да се управляват потребителски акаунти, скриптът задава поредица от въпроси, а съответните отговори определят промените, извършвани във файла passwd. Скриптовете в UNIX/Linux са еквиваленти на съветниците в Windows.

4.3 Директорийни услуги

Една важна част от модерните мрежови операционни системи (network operating systems — NOS) е директорийната услуга.

Директорийната услуга (directory service) позволява съхраняването и осъществяването на достъп до информацията относно мрежови ресурси, мрежови акаунти и мрежови услуги.

Мрежовите операционни системи, които използват директорийни услуги изискват два компонента: самата директория и услуга, която я управлява.

Една *директория* организира информацията в нея по такъв начин, че да се представи на потребителя в подреден и изискван от него вид. Абонатите в една мрежа ежедневно използват директории (справочници). В света на компютрите, една директория (справочник) може да е много различни неща.

Както е известно файловете системи използват това понятие, като директорията представлява сбор от файлове, групирани под едно идентифициращо ги име.

Когато обаче се говори за директорийните услуги, използвани от мрежовите операционни системи, става въпрос за различна концепция. В този контекст директорията (справочникът) е специален тип база данни. Тя може да съдържа различни типове информация. Често в терминологията на компютрите, под името директория се има предвид справочник.

В една обектно-ориентирана операционна система директорията съдържа *обекти*, които имат *атрибути*. Например един потребителски акаунт, представлява обект, който може да се съдържа в директория. Неговите атрибути ще включват потребителското име, истинското име на потребителя, паролата за акаунта и друга информация за акаунта, както е показано на фиг.4.8.

Директорийните услуги предоставят начин за съхранение, обновяване, откриване и защита на информацията в директорията. Директорийните услуги могат да бъдат *локални* (т.е., ограничени до една машина) или *глобални* (т.е. да осигуряват услуги на множество машини). Ако самата информация е разпръсната на множество машини, тя се нарича *дистрибутирана*.

Използване на директорийни услуги

Ползите от използването на директорийни услуги в една мрежа включват следното:

- Данните могат лесно да бъдат организирани.
- Данните могат лесно да бъдат обезопасени.
- Данните могат лесно да бъдат откривани и да се осъществява достъп до тях.

За да се разбере ползата от използването на директорийни услуги, ще бъде пояснен традиционния начин за осъществяване на достъп до мрежови ресурси.

При традиционния начин, споделените файлове и папки се съхраняват на твърдите дискове на отделните работни станции или файлови сървъри. За да се свърже със споделения ресурс, потребителят трябва да знае неговото местоположение.

Една директорийна услуга елиминира това изискване. Споделените ресурси се публикуват в директорията, а потребителите могат да ги откриват и да осъществяват достъп до тях, без изобщо да знаят на коя машина ресурсът е разположен физически.

Популярни директорийни услуги

Двете най-популярни директорийни услуги за PC мрежи са Novell NDS и Microsoft Active Directory.

Структура на базата данни NDS

Започвайки с версия 4, NetWare въведе NDS - глобална база данни, която се репликира между сървърите в мрежата. Чрез нея потребителите могат да се логнат, в който и да е сървър, след което да осъществяват достъп до ресурсите на всички сървъри.

Базата данни NDS е йерархична и използва схемата с инверсно дърво. Тя може да съдържа два основни типа обекти: *контейнерни обекти (container objects)* и *листови обекти (leaf objects)*. Както се подразбира от имената, един контейнерен обект може да съдържа други обекти в себе си; листовият обект е „крайната точка“ на даден клон - самият ресурс. Споделените файлове и принтери са примери за листови обекти. OU единиците са примери за контейнерни обекти.

Microsoft Active Directory

С пускането на пазара на Windows 2000 Server, Microsoft направи фундаментални промени в своите мрежови компоненти, които са дори по-драстични от промените, извършени от Novell при преминаването от NetWare 3 към 4. Active Directory заема централно място в тези промени. Докато NDS

базата данни на Novell действа като услуга, работеща с мрежовата операционна система, Active Directory на Microsoft действа като приложение, което е дълбоко интегрирано в операционната система.

Структура на базата данни на Active Directory

Информацията в Active Directory се съхранява в три файла:

- Active Directory база данни
- Active Directory отчетни файлове
- Shared System Volume

Самата база данни представлява директорията. Отчетните файлове записват промените, извършени в базата данни. Shared System Volume (наричан още Sysvol) съдържа скриптове и *обекти на групови политики (group policy objects)* на Windows 2000 домейн контролери. *Груповата политика* е средството, чрез което Windows 2000 администраторите контролират десктопите на потребителите, автоматично инсталират приложения и установяват правата на потребителите.

Active Directory u DNS

Active Directory използва DNS конвенциите за именуване и зависи от DNS, за да работи. Трябва да има DNS сървър във всяка Windows 2000 мрежа. Освен това обновяванията на информацията за DNS зоните може да се интегрира с репликирането на Active Directory, което е по-ефективно от традиционните методи за обновяване на DNS.

Windows 2000 поддържа Dynamic DNS (DDNS), което позволява автоматично обновяване на DNS базата данни.

Сигурност в Active Directory

Всеки обект в Active Directory притежава списък за контрол на достъпа (access control list - ACL), който съдържа всички разрешения, асоциирани с обекта. Разрешенията могат изрично да бъдат давани и отказвани на грануларен принцип.

Съществуват два типа разрешения:

- **Назначени разрешения** - Разрешения дадени изрично от потребител, който има властта да направи това
- **Наследени разрешения** - Разрешения, които се прилагат върху дъщерни обекти, тъй като са наследени от родителски обект

Разрешения могат да се назначават на отделен потребител или група от потребители.

Windows 2000 позволява на администраторите да контролират процеса на наследяване, като по този начин при желание могат да предотвратят наследяването. В случая е налично поле за отметка със свойствата за сигурност на обекта.

4.4 Политики за сигурност

Защитата на данните в мрежата често изисква прилагане на комбинирани методи за сигурност.

Нива на сигурност на операционната система

Операционните системи с висока сигурност, като Windows NT, Windows XP и Linux, изискват въвеждането на валидно потребителско име и парола, за да се зареди операционната система и тя да бъде използвана. Също така тези операционни системи съхраняват паролите в криптирана форма, до която не може да бъде осъществен лесен достъп.

Потребители, групи и разрешения

Модерните операционни системи позволяват на множество потребители да осъществяват достъп до компютъра и мрежата чрез създаване на отделни потребителски акаунти, и дават възможност за задаване на различна парола за всеки от тях. Въвеждането на комбинацията от потребителско име и парола от промпта за логване води до следния резултат:

- На потребителя се дава достъп до операционната система и мрежата.
- Потребителят може да чете и записва в тези споделени ресурси, за които на неговия акаунт са дадени разрешения.
- Потребителят може да упражнява такива права (например правото да изключва операционната система или да инсталира програми), каквито са зададени на неговия акаунт.
- Зареждат се общите настройки за потребителя, като например икони на десктопа и тапет (wallpaper).

В някои компютърни среди, в които на компютрите не се съхраняват конфиденциални данни и няма изисквания за сигурност, може да не се създават отделни потребителски акаунти. Всички потребители могат да използват един и същ акаунт за логване. Това означава, че всеки потребител има едни и същи разрешения за достъп. Това не само създава отворена, небезопасна среда, а също означава, че потребителите не могат да приспособяват по свой вкус десктопа и други настройки.

Политики за контрол на достъпа

На всеки отделен потребител може да бъдат дадени точно позволенията, от които се нуждае. С помощта на файлови системи с висока сигурност позволенията могат да бъдат задавани не само за ресурси, до които се осъществява достъп по мрежата, а също и за ресурсите, до които се извършва достъп от самата локална машина. Локалните и мрежовите разрешения не е задължително да бъдат едни и същи.

Важно е да се знаят подразбиращите се разрешения, използвани от различните мрежови операционни системи. Например при Windows NT или Windows 2000 Server, когато се сподели ресурс, той е напълно достъпен за всеки в мрежата, докато не се отменят изрично тези разрешения. На сървър с NetWare е в сила обратното - споделеният ресурс не е достъпен за никого чак докато изрично не се променят позволенията на даден потребител.

Нито един от методите не е изцяло правилен или погрешен, но ако не знаете как работят позволенията на вашата мрежова операционна система, може да се окаже, че е разрешен достъп до ресурси, които не е трябвало да бъдат открити, или пък е забранен достъпа на потребители, които трябва да имат такъв.

В организация, заинтересувана от прилагане на система за сигурност, трябва да бъдат създадени политики, в които да бъде указано кой трябва да има достъп и до кои ресурси. Най-общо достъпът трябва да бъде предоставян на принципа „трябва да знае“. Ако потребителят трябва да има достъп до ресурса, за да изпълнява своите служебни задачи, трябва да бъдат дадени разрешения. В противен случай достъпът трябва да бъде отказан.

Използване на групи на сигурност

Групите на сигурност са създадени в помощ на администраторите и се поддържат от много мрежови операционни системи с цел улесняване на задаването на разрешения в голяма мрежа. Създават се групи, и на тези групи се дават разрешения за достъп до ресурси. След това в групата се поставят съответните потребителски акаунти, като по този начин на всеки потребител веднага се дават всички разрешения, зададени за групата като цяло. Това е лесно, отколкото задаването на разрешения на всеки потребител поотделно.

Ако даден потребител не трябва да има достъп до определен ресурс, тогава е достатъчно той да бъде премахнат от членство на всяка една група.

Криптиране на файлове

Криптирането включва конвертиране на данните във форма, която не може лесно да бъде разбрана от другите.

Криптирането на файлове е способ за криптиране на данни, съхранявани на диска на компютъра, така че те да не могат да бъдат прочетени от никого, а само от създателя на данните. Някои операционни системи, като Windows 2000, включват функция за криптиране на файлове. За тези, които не поддържат такава функция са достъпни програми за криптиране от външни производители.

Когато документите са криптирани на диска, може да ги разглежда само потребител, който има правилен ключ. Ако други опитат да осъществят достъп, тогава или файлът няма да се отвори изобщо, или ще се появи като объркани безсмислени знакове.

Необходимо е да се обърне внимание на това, че конфиденциалните данни трябва да бъдат защитени както с разрешения за достъп, така и с криптиране.

Груповите политики упростяват администрирането, като осигуряват на мрежовия администратор централизиран контрол върху привилегиите, правата и възможностите на потребителите компютрите.

Груповата политика представлява набор от правила, които подпомагат управлението на компютри и потребители.

Груповите политики могат да бъдат приложени за множество домейни, отделни домейни, подгрупи от даден домейн или отделни системи.

Политиките, които се прилагат на отделни системи носят наименованието локални групови политики и се съхраняват само на локалната система.

Останалите групови политики могат да бъдат свързани като обекти в услугата Active Directory.

Управлението на политиките е изключително важно за правилната работа на отделните системи в мрежата. Именно заради това те трябва да се използват

много внимателно, защото некоректната им дефиниция може да изолира дадена система от ползването на ресурс в мрежата.

Политиките се разделят основно на две големи групи:

- **компютърни**
- **потребителски**

Компютърните политики са валидни за дадена система и се прилагат когато компютъра се стартира.

Потребителските политики се прилагат когато потребителя влиза в компютърната система.

Конфигуриране на политиките

Когато администраторът иска да управлява потребители и компютри, може да конфигурира политики, достъпни чрез административни шаблони (*administrative templates*). Тези политики осигуряват лесен достъп до настройки от Регистъра, които контролират операционната система, компонентите на Windows и програмите.

Разглеждане на политиките и шаблоните

По принцип конфигурираните в момента шаблони могат да бъдат разглеждани чрез възела Administrative Templates на конзолата Group Policy. Този възел съдържа политики, които могат да бъдат конфигурирани за локални системи, организационни единици, домейни и сайтове. Под Computer Configuration и User Configuration има различни набори от шаблони. Тук може ръчно да се добавят допълнителни шаблони, съдържащи нови политики - или чрез конзолата Group Policy, или когато се инсталират нови компоненти на Windows.

Промените в политиките, достъпни чрез административни шаблони, се записват в Регистъра. Компютърните конфигурации се записват в HKEY_LOCAL_MACHINE, а потребителските - в HKEY_CURRENT_USER. Най-добрият начин да се разбере с какви политики, достъпни чрез административни шаблони, се разполагате, е да се разгледа възела Administrative Templates в конзолата Group Policy.

Политиките могат да бъдат в едно от следните три състояния:

- **Not Configured** Политиката не се използва и в Регистъра няма записани настройки за нея.
- **Enabled** Политиката активно се прилага и нейните настройки са записани в Регистъра.
- **Disabled** Политиката е изключена и не се прилага. Тази настройка се записва в Регистъра.

4.5 Дискови квоти

Дисковите квоти се използват от администраторите за да се управлява заетото дисково пространство на критични за мрежата дялове на харддиска (томове). Такива могат да бъдат толове, които съдържат споделени папки както с корпоративни данни, така и с данни на съответните потребители в мрежата.

За целта е необходимо да бъде създаден лимит на дискова квота и съответно предупредително ниво на тази квота. Лимитът на дисковата квота определя горната граница на използваното дисково пространство. Ако се достигне този лимит има възможност да бъде забранено използването на допълнително пространство от дадения том съответния потребител. От друга страна е възможно настъпването на това събитие да бъде регистрирано в дневник (log файл).

Предупредителното ниво на дисковата квота се използва за предупреждение на потребителите за приближаване на линията за дисковата квота и съответно записване на това събитие в дневник.

По подразбиране дисковите квоти и тяхното управление е забранено и е необходимо те да бъдат разрешени

В повечето случаи се дефинират т.нар. задължителни квоти, при които за всеки потребител се определя неговия лимит и при неговото достигане се отказва използването на повече дисково пространство.

Най-добрите практики показват, че е желателно създаването на допълнителни незадължителни квоти. При тях е необходимо да бъде следено заетото дисково пространство на отделните потребители и администратора да бъде информиран за това кога е надвишен някакъв предварително дефиниран лимит, вместо да бъде отказвано допълнително дисково пространство на потребителите. Очевидно тази информираност може да стане чрез записване на настъпилото събитие (превишаването на определения лимит) в дневник, който администратора може да преглежда.

Дисковите квоти се прилагат само за крайни потребители, но не и за администраторите.

На администраторите не може да бъде отказвано дисково пространство, дори и при надвишаване на техния лимит, наложени от дадена дискова квота. В повечето случаи дисковите квоти се задават в MB (Mega Bytes) и GB (Giga Bytes).

4.6. Защита на данните и избягване на сривове

4.6.1 Физическа сигурност

Политиката на мрежова сигурност трябва да контролира *физическия достъп* до компонентите на мрежата.

Това е важен, но често пренебрегван фактор при разработване на ефективен план на сигурност.

За целта трябва да бъде определена степеня на директен физически достъп, който служителите, предприемачите, клиентите и обществеността ще имат до работните станции, сървърите, кабела или друга преносна среда, маршрутизаторите, суичовете и другите физически компоненти.

В среда с висока степен на сигурност сървърите и устройствата за връзка трябва да се пазят зад заключени врати. Работните станции, които се намират в незащитени области трябва да имат софтуерни средства за контрол, които не допускат достъп до конфиденциални данни в мрежата.

В много организации се полагат големи усилия за ограничаване на достъпа на служителите и обществеността до компютърното оборудване - и след това цялото оборудване се оставя широко отворено всяка нощ, когато идва обслужващият персонал за почистване.

Ако сигурността е важен въпрос за разработваната мрежа, персоналят по почистването и поддръжката трябва да бъде под наблюдението на оторизирано лице, когато се почистват стаите, съдържащи мрежово оборудване.

4.6.2 Защита и възстановяване от сринове

Външните и вътрешните нарушители не са единствената заплаха за мрежата и данните в нея.

Също така могат да възникнат хардуерни повреди, природни бедствия и технически грешки, които също довеждат до опустошителни загуби на важни файлове.

Ето защо мерките за защита и възстановяване от сринове са важна част от всяка производствена мрежа.

Защитата срещу катастрофални загуби на данни и възстановяването на данните след такива сринове включва няколко линии на защита:

- Аварийно захранване
- Архивиране на данните
- Отказоустойчивост при дисковете
- Отказоустойчивост на ниво сървъри (клъстериране)

4.6.3 Архивиране на данните

Независимо от усилията за предотвратяване на проблеми със захранването, винаги е възможно да се случи нещо такова: да настъпи отказ на твърд диск, пожар или наводнение да унищожат сървъра, или злобен вирус да форматира устройството и да направи данните нечетими. Просто данните ще бъдат изгубени.

Но ако бъде реализирано редовно и изчерпателно архивиране на данните с помощта на програма за архивиране, те не са изгубени завинаги.

Създаването на план за архивиране включва редица въпроси.

Планът за архивиране може да отчита разликата между катастрофална загуба на данни, време и пари, и малкото неудобство от отделянето на няколко часа за възстановяване на файловете от архива до тяхното първоначално състояние.

В повечето случаи графикът за архивиране включва различни типове архивиране. Има три основни типа (макар че някои помощни програми за архивиране ви дават допълнителни възможности за избор):

- **Пълно архивиране** - Всички данни на зададените устройства се архивират независимо от това, дали и кога са архивирани преди, и дали са променени след последното архивиране.
- **Диференциално архивиране** - Архивират се всички файлове, променени след последното *пълно* архивиране.
- **Инкрементално архивиране** - Архивират се всички файлове,

променени след последното архивиране от всеки тип (а не просто след последното пълно архивиране).

4.6.4 Отказоустойчивост при дисковете

Друг начин за защита на данните от повреда на твърдия диск е да реализирате *отказоустойчивост*. Отказоустойчивостта означава възможността на дадена система да се възстановява след срив.

Това включва комбиниране на множество физически твърди дискове в един *отказоустойчив пакет*, който може да бъде в една от няколко възможни конфигурации.

Отказоустойчивостта при дисковете се нарича също редундантен масив от независими (или нескъпи) дискове *{redundant array of independent (inexpensive) disks - RAID}*).

Най-често използваните отказоустойчиви конфигурации са следните:

- **Дублиране на диск или създаване на огледално копие на диск (RAID level 1)** - За дублиране на диска са необходими два физически твърди диска, за предпочитане с един и същ капацитет. Всички данни от единия диск се дублират на втория. На втория диск има точно дублирано копие на всички файлове и структури. Ако единият диск се повреди, другият може да поеме работата. Това става или автоматично, или може да се наложи да се укаже на операционната система къде да намери новия диск.

Например в Windows NT това се прави чрез редактиране на файла boot.ini.

- **Дуплексиране на диск (RAID level 1)** - Дуплексирането на диск работи по същия начин като дублирането, с изключение на това, че двете физически устройства се свързват към различни дискови контролери. Това повишава нивото на отказоустойчивост, защото ако контролерът на единия диск се повреди, другият диск продължава да функционира, тъй като се управлява от различен контролер.

- **Лентов запис (страйпинг) с дисково устройство за четност (RAID level 3)** - Този метод за конфигуриране се състои в записване на данните на ленти едновременно на множество еднакви устройства и записване на данните за самите записи върху тези устройства на друго устройство, което е запазено само за тази цел. Това изисква минимум три физически диска (два еднакви диска, на които данните се записват на еднакви ленти, и един за информацията за четност). Ако се повреди диск с данни, данните могат да бъдат регенерирани с помощта на информацията за четност.

- **Лентов запис (страйпинг) с ленти за четност (RAID level 5)** - Лентовият запис с ленти за четност работи по начин, подобен на RAID 3, с изключение на това, че четността (подобно на данните) се записва в ленти, които се променят между дисковете. Нито един диск не е предназначен за четност. И тук са необходими поне три диска, но данните могат да бъдат възстановени при повреда на *който и да е* от тях.

RAID може да бъде реализиран или софтуерно, или хардуерно. Макар че хардуерно базираният RAID е по-бърз и като цяло по-надежден, той е и по-

скъп. Някои сървърни операционни системи, като Windows NT и Windows 2000 Server, имат вградена поддръжка за софтуерен RAID.

Глава пета

Технологии на сигурността в мрежите

Съществуват много технологии за сигурност, предоставящи решения за защита на мрежовия достъп и механизмите за пренос на данни в една мрежова инфраструктура. Много от технологиите се припокриват в разрешаването на проблеми, отнасящи се до осигуряване на идентичност на потребител или устройство и поверителност на данните.

От тук нататък понятията *удостоверяване (authentication)*, *оторизация (authorization)* и *контрол на достъпа (access control)* са обединени в концепцията за идентичност. Въпреки че тези концепции са различни, всички те се отнасят за отделния потребител в мрежата - независимо дали е човек, или устройство. Всеки човек или устройство е отделен обект, притежаващ различни възможности в мрежата, на когото се позволява достъп до ресурси на базата на това, кой е той. Въпреки че в буквален смисъл, идентичността се отнася само за удостоверяването, в много случаи е разумно да разглеждаме едновременно и оторизацията и контрола на достъпа на обектите.

Удостоверяването се отнася за процеса на валидиране на заявената идентичност на краен потребител или устройство (например клиенти, сървъри, кому-татори, маршрутизатори, защитни стени и т.н.).

Оторизацията се отнася за процеса на предоставяне на права за достъп на потребител, група потребители или дадена система; *контролът на достъпа* се отнася за ограничаване на потока информация от ресурсите на дадена система само към оторизираните лица или системи в мрежата. В повечето случаи, оторторизацията и контролът на достъпа във времето са след успешно удостоверяване.

В тази глава се описват технологии за сигурност, използвани често за установяване на идентичност (удостоверяване, оторизация и контрол на достъпа) и за осигуряване на някаква степен на цялост на данните и поверителност в мрежата. *Целостта (integrity)* на данните осигурява засичането на всяка промяна или повреда на данните от хора, които не са изрично предвидени да правят това; *поверителността (confidentiality)* на данните осигурява виждането на данните в използваем формат само от страните, на които това е позволено.

5.1 Основни криптографски концепции

Технологиите за сигурността на компютрите се базират на науката *криптография*, която е учение за „тайнописа“ или *шифъра*. Криптирането използва код или ключ за разбъркване и след това за подреждане (или *дешифриране*) на съобщението, с цел връщането му в неговата първоначална форма. Задачата по анализиране и дешифриране на криптираните съобщения се нарича *криптоанализ*.

Данните се криптират с помощта на *алгоритъм* или шифър. Криптираните данни се означават като *шифрован текст*.

Целта на криптирането е чрез използване на „секретните кодове“ да се защити частния характер на комуникацията между два обекта.

Една много проста схема на криптиране задава число на всяка буква от азбуката. Ако се представи буквата А като 1, В като 2 и т.н., можем да бъде изпратено „кодирано“ съобщение като следното:

4-9-14-14-5-18 9-19 18-5-1-4-25

Съобщението лесно може да бъде „декодирано“ или *декриптирано* до оригинала:

Dinner is ready (вечерята е готова)

За нещастие, този шифър лесно може да бъде разбит (кракнат) и затова не е сигурен.

Криптирането при компютрите е по-сложно.

Най-общо се използва *ключ (key)*, който е някакъв вид променлива, комбинираща се с данните, които се криптират. Методът на комбиниране на ключа с данните се нарича *алгоритъм*. Тъй като компютрите обработват двоична информация, тези изчисления се прилагат към всеки бит или група от битове.

Колкото по-дълъг е ключът за криптиране, толкова по-трудно е да се разбие кодът. В областта на криптирането 40- и 56-битовото криптиране се нарича стандартно криптиране, а 128-битовото криптиране се нарича силно криптиране.

Съществуват два популярни типа криптиране: криптирането със секретен ключ и криптирането с публичен/частен ключ.

Криптиране със секретен ключ

Криптирането със секретен ключ често се означава като *симетрично криптиране*, защото един ключ се използва както за криптиране, така и за декриптиране на данните. Предаващата и приемащата страни се договарят за използване на общ ключ или криптиращ алгоритъм, който представлява *споделена тайна*.

Криптиране с публичен/частен ключ

Макар че за краткост често пъти се означава като криптиране с публичен ключ, по-точният термин е *криптиране с публичен/частен ключ (public/private key encryption)*, тъй като този тип криптиране използва два ключа - единият от тях се публикува и е широко достъпен, а другият е частен и се знае само от потребителя. За извършване на сигурната комуникация са необходими и двата ключа. Този тип криптиране се нарича също *асиметрично криптиране*. При него всеки потребител има и публичен, и частен ключ, наречени *двойка ключове*.

За да работи този вид криптиране, трябва да бъдат използвани и двата ключа на една двойка ключове; не е нужно никой да знае частния ключ на някой друг.

Автентикация

Примерите, които бяха разгледани дотук, се занимават с *конфиденциалността* на данните. Отделен проблем е *автентикацията* на изпращащия данните.

С помощта на метода на публичния/частния ключ може лесно да се реализира и осигури сигурността за самите данни. Но тъй като публичният ключ е достъпен за всеки, Иван не може да знае със сигурност, че съобщението, което е приел и което е криптирано с неговия публичен ключ, наистина е от Мария. За да провери истинността на самоличността на изпращащия (да го автентичира), Мария трябва да криптира съобщението с нейния *частен* ключ. След това Иван може да го декриптира с нейния публичен ключ, като е сигурен, че тя е истинският подател, защото само тя знае собствения си частен ключ.

Друг начин за осигуряване на автентикация на изпращащия е да се използват цифрови подписи.

Цифрови подписи

Цифровите подписи се състоят от криптирана подписваща информация, добавена към документа. Тази информация верифицира както идентичността на изпращащия, така и целостта на самия документ, цифровите подписи не проверяват данните. Те само гарантират, че данните не са променени и че изпращачът е автентичен.

Алгоритмите с публичен ключ се използват за създаване и верифициране на цифрови подписи и *хеш алгоритми*.

Хеш алгоритми

Хешът представлява резултатът от еднопосочно математическо изчисление (хеширащ алгоритъм), което създава *дайджест на съобщение* (*message digest*). Алгоритъмът се нарича „еднопосочен“, защото не може да се обръща (реверсира) резултата, за да се открие оригиналното съобщение.

Цифрови сертификати

Цифровите сертификати представляват съобщения, които съдържат цифровия подпис на доверена трета страна, наречена *сертификационна власт* (*certificate authority*). Третата страна гарантира, че конкретен публичен ключ реално принадлежи на конкретно лице. Сертификатите се използват за гарантиране на автентичността на съобщенията, които пътуват по несигурни публични мрежи, например Интернет.

Технологии за идентичност

Удостоверяването на идентичност на хост, краен потребител или на двете е изключително важен елемент. В много корпоративни мрежи няма да се предостави оторизиран достъп до определени части от мрежата преди да се установи кой се опитва да получи достъп до тези поверителни източници. Кой кого трябва да идентифицира, е сериозен въпрос. В някои случаи, инициаторът на комуникацията е страната, задължена да се удостовери, а в други е отговарящият. Понякога се изисква взаимна удостоверяване. За да се усложнят нещата допълнително, стои въпросът и дали да се удостовери само крайното устройство или и същинския потребител, и евентуално дали те да бъдат асоциирани един с друг. В много от случаите, множество крайни потребители

използват едно и също устройство, а може да имат нужда от различни права в мрежата. Освен това разпространението на преносимите компютри (лаптопи) улеснява хората да пътуват и да се свързват с корпоративните офиси от всяка точка на света. Ако някой лаптоп бъде откраднат и той използва автоматизирана схема за удостоверяване и права за достъп, базирана само на идентичността на устройството, това ще предизвика множество усложнения за сигурността.

Методите за удостоверяване могат най-общо да бъдат категоризирани като такива, при които има *локален контрол (local control)*, и такива, при които се предоставя проверка на удостоверяването посредством доверена *трета страна (trusted third party)*.

Протоколът PPP

PPP е стандартизирано Интернет капсуловане на IP в point-to-point връзки. PPP се отнася към такива проблеми като присвояване и упразнение на IP адреси, асинхронно (старт/стоп) и битово-ориентирано синхронно капсуловане, мултиплексиране на мрежовия протокол, конфигурация на връзката, тестване на качеството на връзката, засичане на грешки и договаряне на опции за възможности като договаряне на мрежови адрес на слой и договаряне на компресия на данните. PPP се справя с тези проблеми като предоставя разширяем Link Control Protocol (LCP) и семейство от Network Control Protocols (NCPs) за договаряне на незадължителни параметри и улеснения на конфигурацията. След като връзката бъде установена, PPP дава възможност за още една, незадължителна, фаза на удостоверяване преди да премине към фазата на протокола за мрежовия слой.

PPP канален слой

Частта за данни на протокола PPP (PDR) използва рамката High-Level Data Link Control (HDLC), определена в ISO 3309-1979 (и допълнена от ISO 3309-1984/PDAD1).

PPP договарянето се състои от LCP и NCP договаряне. LCP е отговорен за установяване на връзка чрез определени договорени опции, поддържане на връзката и предоставяне на процедури за прекъсването ѝ. За да изпълни тези функции, LCP е организиран в следните четири фази:

- 1 Установяване на връзката и договаряне на конфигурацията
- 2 Определяне на качеството на връзката
- 3 Договаряне на конфигурацията на протокола за каналния слой
- 4 Прекъсване на връзката

За да се установи комуникация през point-to-point връзка, всеки край на PPP връзката първо трябва да изпрати LCP пакети, за да конфигурира връзката за данни по време на фазата на установяване на връзката. След като връзката бъде установена, PPP предоставя възможност за допълнителна фаза на удостоверяване преди да премине към фазата на протокола за мрежовия слой. NCP фазата след това установява и конфигурира различни протоколи за мрежовия слой, например IP.

По принцип удостоверяването преди NCP фазата не е задължително. Ако се изисква удостоверяване на връзката, имплементацията ще определи конфигурацията на протокола за удостоверяване по време на фазата на установяване на връзката. Тези протоколи за удостоверяване са предвидени за употреба предимно при хостове и маршрутизатори, свързващи се към сървър от PPP мрежа посредством комутатори или dialup линии, но могат да се прилагат и при специализирани връзки. Сървърът може да използва идентификацията на свързващия се хост или маршрутизатор в селекцията от опции за договаряния в мрежовия слой.

PPP протокол за удостоверяване на пароли

Password Authentication Protocol (PAP) предоставя прост начин на едната страна да укаже идентичността си на удостоверяващия посредством двустранно ръкостискане. Това се прави само при първоначалното установяване на връзката.

След като бъде приключена фазата на установяване на връзката, пакетът за заявка на удостоверяване се използва за започване на PAP удостоверяването. Този пакет съдържа името и паролата на едната страна.

PAP не е силен метод за удостоверяване. PAP удостоверява само едната страна, а паролите се пращат в чист текст. PAP удостоверяване се извършва само по веднъж на сесия. Няма защита от риплей атаки или от повторяеми атаки от типа „проба и грешка“. Удостоверяваният управлява честотата и продължителността на опитите.

PPP Challenge Handshake Authentication Protocol

Challenge Handshake Authentication Protocol (CHAP), дефиниран в RFC 1994, се използва за периодично потвърждаване на идентичността на хост или краен потребител посредством трипосочно ръкостискане. CHAP се извършва в началното установяване на връзката и може да бъде повторен по всяко време след като връзката е установена.

CHAP налага мрежова сигурност като задължава страните да споделят обща тайна фраза в чист текст. Тази тайна фраза никога не се изпраща по връзката.

Тайните пароли трябва да бъдат еднакви в отдалеченото и локалното устройство. Тези тайни трябва да са договорени, генерирани и обменени по сигурен начин. Тъй като тайната фраза никога не се предава, това предотвратява открадването ѝ от други устройства и получаването на неправомерен достъп до системата. Без правилния отговор, отдалеченото устройство не може да се свърже с локалното.

CHAP предоставя защита срещу плейбек атаки посредством използването на нарастващ идентификатор и променлива стойност на поканата. Употребата на повтарящи се покани е с цел намаляване на времето на уязвимост към атака. Удостоверяващият управлява честотата и времетраенето на поканите.

Всеки от двете CHAP страни може да играе ролята на удостоверяващ в спецификацията няма изискване удостоверяването да бъде пълен дуплекс или в двете посоки да се използва един и същ протокол.

PPP Extensible Authentication Protocol

PPP *Extensible Authentication Protocol* (EAP) е общ протокол за PPP удостоверяване, поддържащ множество механизми. Той предоставя собствена поддръжка на елиминация и препредаване на дублирането. Самият EAP не поддържа фрагментиране, но отделни EAP методи може да го поддържат.

EAP не избира определен механизъм за удостоверяване през фазата за управление на връзката; вместо това го отлага за фазата на удостоверяването, така че удостоверяващата страна да има възможност да поиска повече информация преди да определи конкретния механизъм за удостоверяване. Такава подредба позволява, също така, използването на бек-енд сървър, който всъщност имплементира различните механизми за удостоверяване, докато PPP удостоверяващият само препредава информацията по удостоверяването. Така се получава и предимството да не е необходимо удостоверяващият да бъде обновяван, за да поддържа всеки нов метод за удостоверяване.

EAP добавя гъвкавост към PPP удостоверяването и предоставя възможност за използване на нови технологии - например цифрови сертификати.

Протоколи, използващи механизми за удостоверяване

Много протоколи изискват проверка на удостоверяването преди да предоставят оторизация и права за достъп на потребителя или устройството. Такива протоколи са TACACS+, RADIUS, Kerberos, DCE и FORTEZZA. TACACS+ и RADIUS често се използват dial-in среди за предоставяне на мащабируема база данни за удостоверяване и могат да обединяват различни методи за удостоверяване. Kerberos е протокол, използван в някои университетски среди, който потвърждава предварително, че потребителите и мрежовите услуги, които те използват, са тези, които твърдят, преди да им даде права за достъп.

5.2 Протоколи за сигурност на приложния слой

Приложният слой се отнася за определено приложение, като Telnet, FTP или HTTP, а не за подробностите относно преноса на данни по мрежата. Този слой използва end-to-end протоколи, при които крайните системи са отговорни за предоставяне на сигурност за приложния протокол. Не много протоколи за сигурност са проектирани специално за отделни приложения. Съществуват прекалено много приложения, за да бъде подобен подход мащабируем. Някои обаче заслужават да бъдат споменати. Тъй като WWW се е превърнало в едно от най-бързо разрастващите се приложения в Интернет, беше проектиран специален протокол за сигурност за защита на уеб транзакциите: Secure HyperText Transport Protocol (SHTTP). Протоколът Secure Multipurpose Internet Mail Extensions (S/MIME) беше проектиран за добавяне на функционалност за защита към протокола MIME, за да бъде интегриран в продукти за е-поща и съобщения.

5.3 Протоколи за сигурност в транспортния слой

Транспортният слой предоставя подробности за движението на потока от данни между два хоста. *Протоколи за сигурност на транспортния слой*, имат за целт да защитят транспортния слой и да предоставят методи за имплементация на поверителност, удостоверяване и цялост над транспортния слой. Този слой използва end-to-end протоколи, при които крайните системи са отговорни за предоставяне на защита на транспортния протокол.

Протоколът Secure Socket Layer/Transport Layer Security

Протоколът Secure Socket Layer (SSL)/Transport Layer Security (TLS) определя механизъм за предоставяне на защита на данните, разпределен между приложни протоколи (като HTTP, Telnet, NNTP или FTP) и TCP/IP. Той предлага криптиране на данните, удостоверяване на сървъри, цялост на съобщенията, и евентуално, удостоверяване на клиента при TCP/IP връзка.

SSL предполага, че намиращият се по-ниско механизъм за доставка на пакети, е надежден, и въпреки че на теория съществуват редица протоколи, които биха предоставили тази услуга, SSL почти винаги използва TCP за транспорт.

Основната цел на SSL е да предостави поверителност и надеждност между две общуващи приложения. SSL е протокол със слоеве, състоящ се от протокола за записи, предоставящ опаковката, и услуги за сигурност за четирите протокола за слоя данни (handshake, alert, change cipher spec и application).

SSL/TLS е широко използван в HTTP трафика. Други протоколи също започват да използват SSL/TLS и за тях се дефинират специални номера на портове Таблица 5.2.

Протоколът Secure Shell

Secure Shell (SSH) е протокол за защитено отдалечено влизане (login) и други защитени мрежови услуги през незащитена мрежа. Той предоставя поддръжка за защитено отдалечено влизане, защитен пренос на файлове и защитено препредаване на TCP/IP и X Window System трафик. Той има възможност за автоматично криптиране, удостоверяване и компресиране на предаваните данни. Работният процес по дефинирането на протокола SSH осигурява протоколът SSH да предоставя солидна защита срещу криптоанализ и протоколни атаки, да може да работи достатъчно добре без глобална инфраструктура за управление на ключове или сертификати, и да може да използва съществуващите инфраструктури за сертификати

Протоколът SOCKS

Socket Security (SOCKS) е защитен мрежов прокси протокол, базиран на транспортния слой. Той е проектиран за предоставяне на рамка за клиент/сървър приложения в TCP и UDP среди за удобно и сигурно използване на услугите на мрежова защитна стена.

5.4 Защита на мрежовия слой

Защитата на мрежовия слой спада към услугите по сигурността от нивото на IP от протоколния стек TCP/IP. Мрежовият слой предоставя постъпкова

обработка на пакети с данни, при което може да участват и междинни системи от мрежата, като например маршрутизатори. Във всяка междинна система (стъпка, hop), пакетът данни бива изследван от IP слоя и след това препредаден на следващата междинна система докато се достигне до крайната цел. Много години опит доведоха до разработването на набор от стандарти от страна на IETF, които заедно дефинират как да се защитават услугите на мрежовия IP слой; тези стандарти често се наричат *IPsec*.

Въпреки че проверката на мрежовия слой се извършва на постъпкова база, криптирането/декриптирането на защитени пакети от мрежовия IP слой не е необходимо да става във всяка стъпка.

Комплектът протоколи IP Security

Комплектът протоколи *IP Security* (IPsec) се състои от набор от стандарти, използвани за предоставяне на услуги за поверителност и удостоверяване на IP ниво.

Наборът услуги за сигурност IPsec може да предостави includes контрол на достъпа, цялост без установяване на връзка, удостоверяване на източника на данни, отхвърляне на повторени пакети (вид частична поредна цялост), поверителност (криптиране) и ограничена конфиденциалност на трафика. Тъй като тези услуги се предоставят на IP ниво, те могат да бъдат използвани от всеки протокол от по-високо ниво (като TCP, UDP, ICMP, BGP и т.н.).

Услуги за удостоверяване и криптиране

За да осигури защита на трафика, IPsec използва два протокола, AH и ESP, всеки от които дефинира нов набор от хедъри за добавяне към дейтаграмите на IP.

AH и ESP могат да се използват поотделно или в комбинация, за да се предостави желаните набор от услуги за сигурност. За тези два протокола IPsec не дефинира използването на определени алгоритми за сигурност; вместо това, предоставя отворена рамка за имплементация на алгоритми на индустриални стандарти. Въпреки че някои алгоритми са задължителни за поддръжка, не е задължително точно те да се използват. Общоприето правило е да използвате най-силния алгоритъм, докато не се сблъскате с проблеми с производителността.

Всеки протокол поддържа два режима на работа:

- Транспортен режим
- Тунелен режим

В *транспортен режим*, два хоста предоставят защита предимно за протоколи от високо ниво; криптографските крайни точки (където се извършва криптиране и декриптиране) са източникът и приемникът на пакета с данни. При IPv4, хедърът на протокола за сигурност в транспортен режим се появява веднага след IP хедъра и преди всякакви други протоколи от по-високо ниво (например TCP или UDP).

В случая с AH в транспортен режим, цялата информация от по-горните слоеве е защитена, както и всички полета от IPv4 хедъра с изключение на полетата, които обикновено се променят при преноса.

Тунелът е средство за капсулиране на пакети в протокол, разбираем за входните и изходните точки на дадена мрежа. Тези входни и изходни точки се дефинират като *интерфейси на тунела*. Тунелния режим може да бъде поддържан от крайни точки за пакети с данни и от междинни защитни шлюзове. В *тунелен режим*, „външният“ IP хедър задава IPsec дестинацията за обработка, а „вътрешният“ IP хедър задава крайната дестинация за пакета. Адресът на източника във „външния“ IP хедър е инициращата криптографска крайна точка, която ще бъде променяна с предаването на пакета в мрежата до достигането на целевата криптографска крайна точка; адресът на източника във „вътрешния“ хедър е истинският адрес на източника на пакета. Адресът на дестинацията във „външния“ IP хедър е дестинацията на криптографската крайна точка; адресът на дестинацията във „вътрешния“ хедър е истинският адрес на дестинацията на пакета.

Управление на ключове

IPsec използва криптографски ключове за услуги за удостоверяване/цялост и за криптиране. Поддържа се както ръчно, така и автоматично разпространение на ключове.

Най-ниското (но най-нежелано) ниво от управлението на ключове е *ръчно конфигуриране на ключовете*, при което човек ръчно конфигурира всяка система с ключови данни и данни за SA управление, отнасящи се до сигурната комуникация с други системи. Ръчните техники са практични при малки, статични среди, но не са мащабируеми. Ако броят на сайтовете, използващи услуги за сигурност от IPsec е малък и всички тези сайтове се намират под един и същ административен домейн, техниките за ръчно управление на ключовете може и да са подходящи. Ръчното управление на ключове може да е подходящо и когато трябва да се осигуряват само избрани комуникации в организация с малък брой хостове или шлюзове. Техниките за ръчно управление често използват статично-конфигурирани, симетрични ключове, въпреки че съществуват и други възможности. Избраният за използване с IPsec автоматичен протокол за управление на ключове по подразбиране е Internet Key Management Protocol (IKMP), понякога наричан просто *Internet Key Exchange* (IKE). IKE удостоверява всеки участник в IPsec, договаря политиката за сигурност и поема сигурната обмяна на ключове за сесиите.

5.5 Технологии за сигурност в каналния слой

Технологиите за сигурност при каналния слой се отнасят предимно за тунели. *Виртуалните частни dialup мрежи* (VPDN) дават възможност на големи предприятия да разширяват своите частни мрежи чрез dialup линии. Вместо да имат огромни разходи за осигуряване на сигурност чрез набиране към самото физическо местоположение на фирмата от някъде по света или да намаляват сигурността чрез локално набиране и използването на Интернет, технологиите за тунели в каналния слой дават възможност на отдалечените хостове и потребители да се свързват сигурно с фирмената инфраструктура посредством локален dialup достъп до Интернет.

Понастоящем съществуват три подобни протокола за постигането на тази цел:

- Протоколът Layer 2 Forwarding (L2F)
- Point-to-Point Tunneling Protocol (PPTP)
- Layer 2 Tunneling Protocol (L2TP)

Протоколът Layer 2 Forwarding

Протоколът Layer 2 Forwarding (L2F) е създаден от Cisco Systems. Въпреки че е заместен от L2TP, той е разгледан тук, защото все още се използва в някои среди. L2F разрешава тунелиране през каналния слой — т.е. рамки на High-Level Data Link Control (HDLC), асинхронен HDLC или Serial Line Internet Protocol (SLIP) - на протоколи от по-високо ниво.

Чрез такива тунели е възможно да се раздели местоположението на началния dialup сървър от мястото, на което връзката с dialup протокол се прекъсва и се предоставя достъп до мрежата. Тези тунели дават възможност за приложения, изискващи поддръжка на частно адресирани IP, IPX и AppleTalk dialup, посредством SLIP/PPP в съществуващата инфраструктура за Интернет.

Point-to-Point Tunneling Protocol

Point-to-Point Tunneling Protocol (PPTP) беше започнат от Microsoft и след допълнителна намеса от страна на други доставчици се превърна в информационен стандарт в IETF, RFC 2637. Това е архитектура клиент/ сървър, даваща възможност за тунелиране през IP мрежа и разделяща функциите, съществуващи в настоящите NAS.

Разделяне на традиционната NAS функционалност

Обикновено NAS имплементира следните функции:

Предоставяне на физически собствен интерфейс на PSTN или ISDN мрежи и управление на външни модеми или адаптери за терминали

Предоставяне на логическо прекъсване на PPP LCP сесия Участие в PPP протоколи за удостоверяване

IP тунелът, използващ GRE

PPTP изисква установяването на тунел за всяка комуникираща PNS-PAC двойка. Този тунел се използва за пренос на всички потребителски PPP пакети от сесиите, включващи дадена PNS-PAC двойка. PPTP използва разширена версия, за да пренася потребителските PPP пакети. Тези разширения дават възможност за управление на задръствания на ниско ниво и на потока през тунели, използвани за пренос на потребителски данни между PAC и PNS. Множество сесии биват мултиплексирани през един тунел. Контролната връзка, работеща през TCP, контролира установяването, прекратяването и поддръжката на сесиите, както и на самия тунел.

Layer 2 Tunneling Protocol

Тъй като и L2F, и PPTP предлагат подобна функционалност, Cisco и Microsoft, заедно с други производители, се договориха за единен стандарт, който вече се нарича *Layer 2 Tunneling Protocol* (L2TP). Този протокол е определен в RFC 2661 и често се нарича L2TPv2. Оттогава L2TP се използва за

тунелиране на редица протоколи от други слоеве, а работата продължава за дефиниране на нова версия на протокола, L2TPv3, която ще предоставя по-голяма модулност и по-ясно отделяне от PPP. Тъй като работата по L2TPv3 все още продължава, настоящото разглеждане е насочено към аспектите от работата на L2TPv2.

L2TP задоволява средните изисквания на крайния потребител:

- Прозрачност на крайната система. Нито отдалечената крайна система, нито хостовете ще изискват някакъв специален софтуер, за да използват тази услуга по сигурен начин.
- Удостоверяване, предоставено от dialup PPP CHAP, PAP, EAP или чрез други диалогови прозорци (например текстова обмяна през V.120 преди стартиране на PPP). Това включва TACACS+ и RADIUS решения и поддържа, също така, смарт карти и еднократни пароли. Удостоверяването трябва да бъде управляемо от потребителя, независимо от ISP.
- Адресирането трябва да бъде толкова управляемо, колкото при посветени dialup решения. Адресът трябва да се присвоява от началния сайт, а не от ISP.
- Оторизацията трябва се управлява от началния сайт, както е при решение с директен dialup.
- Отчитането трябва да се извършва едновременно от ISP (за таксуване) и от потребителя (за възстановяване на средства и контрол).

PPPoE

PPP over Ethernet (PPPoE) е дефиниран в RFC 2516 и предоставя средства за капсулиране на PPP пакети в каналния слой. Това дава възможност за свързване на няколко хоста от една подмрежа с концентратор за отдалечен достъп чрез просто мостово устройство за достъп. Този модел се използва предимно в ADSL среди за предоставяне на контрол на достъпа, фактуриране и вид на услугата за всеки отделен потребител, а не за всеки сайт.

PPPoE е с два отделни етапа: етап на откриване и етап на PPP сесия. Етапът на откриване идентифицира Ethernet MAC адреса на концентратора за отдалечен достъп и установява уникален идентификатор на PPPoE сесията. Докато PPP дефинира връзка участник-към-участник, процесът на откриване по същество е клиент/сървър връзка. Могат да бъдат открити много сървъри за достъп, но може да се избере само един. При приключване на успешен процес на откриване, както хостът на клиента, така и избраният концентратор за достъп притежават информацията, която ще използват за изграждане на PPP връзка през Ethernet.

Обобщение

Настоящата глава представи подробно много от настоящите и развиващите се технологии, свързани със сигурността.

Протоколът за сигурност, който се използва в дадена среда, зависи от необходимите услуги за сигурност и от приложенията, които имат нужда от защита. Едно от най-важните съображения е установяването на идентичността на обекта, който иска достъп до корпоративната мрежа. Този процес

обикновено налага удостоверяване на обекта и последваща оторизация и установяване на контрол на достъпа. Някои протоколи са специално проектирани за удостоверяване само на крайни потребители (хора) или само на крайни устройства (хостове, маршрутизатори). Често се налага да комбинирате двата протокола, така че да се удостоверят едновременно крайните потребители и крайните устройства, които те използват за достъп до мрежата.

Защитата в мрежовия слой чрез използването на IPsec има възможност да дефинира услуги за сигурност на IP ниво. В зависимост от имплементациите на производителите, услугите за сигурност могат да бъдат дефинирани да се базират на IP адрес или може да предоставят различни услуги за сигурност, базирани на комбинация от IP адрес, транспортен протокол и приложение. IPsec предлага предимството да скрива информацията от транспортният слой и има възможност да поддържа протоколи от транспортния слой, различни от TCP (например UDP). Тъй като скрива информацията от транспортния слой, ако информацията от хедъра на транспортния слой е необходима за поддръжка на други мрежови изисквания (например QoS, който може да се наложи да погледне номерата на TCP/UDP портовете), може да срещнете проблеми.

Много от протоколите за сигурност изискват или размяната на криптографски ключове, или на цифрови сертификати. PKI трябва да предоставя доверено и ефективно управление на ключове и сертификати. PKI се имплементират или в корпорациите или по по-глобален начин, но специално тази област все още се развива и трябва да се следи внимателно през следващите години.

Глава седма
Качество на услугите в мрежи
Управление на трафик. Приоритизиране на трафик

7.1. Осигуряване качество на услугите

7.1.1 Въведение

Качеството на обслужване (Quality of Service, QoS) се отнася до способността на мрежите да осигурят по добри услуги за определен мрежов трафик провеждан с различни технологии, като Frame Relay (FR), Asynchronous Transfer Mode (ATM), Ethernet и 802.1 мрежи, SONET и IP маршрутизирани мрежи, които могат да използват някои или всички от изброените технологии.

Главната задача на QoS е да се осигури приоритет на даден поток данни посредством заделяне на част от честотна лента, управление на флуктуацията на тактовата синхронизация (джитера) и закъснението за този поток и подобряване характеристиките, зависещи от загубите на пакети. Важно е да се отбележи и осигури, че даването на приоритет на някои от потоците не е за сметка на спирането или провалянето на другите потоци. QoS технологията включва елементни изграждащи блокове, необходими за съставянето на бъдещи бизнес приложения в корпоративни мрежи, WAN и мрежи на доставчици на услуги.

Софтуерните продукти осигуряващи QoS дават възможност на натоварени мрежи да управляват и предвидимо да обслужват разнообразни мрежови приложения и различни типове трафик. Почти всяка компютърна мрежа може да извлече предимства от QoS за оптимална ефективност, независимо дали е малка фирмена мрежа, мрежа на доставчик на Интернет услуги или мрежа на корпорация.

Основните предимства на QoS софтуера са:

- **контрол върху мрежовите ресурси.** Може да се подбере кои ресурси, като четотна лента, оборудване, WAN способности и други могат да се използват. Като пример може да се посочи, че е възможно да се ограничи използваната честотна лента от FTP трафика през гръбначната линия или да се даде приоритет на важния достъп до база данни.

- **по-ефективно използване на мрежовите ресурси.** Това може да стане посредством анализ с помощта на отчитащи инструменти, даващи информация за видовете натоварване на мрежата и възможност за отделяне на по-голямо внимание на важния за бизнеса трафик.

- **услуги при изискване.** Управлението и наблюдението осигурени от QoS дава възможност на доставчика на Интернет услугите да предложи добро тяхно структуриране на потребителите.

- **едновременно използване на критично важните приложения.** QoS технологията трябва да осигури, че WAN се използва ефективно от критично важните приложения, особено необходими за бизнеса. Трябва да се заделят честотната лента и предвиди минималното закъснение изисквани от

чувствителните към тях мултимедии и гласови приложения. Едновременно с това другите приложения, използващи линията трябва да получат благоприятни условия, без да бъдат смущавани от критично важния трафик.

- **трябва да се осигурява основа за напълно интегрирана мрежа в бъдещето.** Използвайки QoS технологията в мрежата се явява добра първа стъпка към напълно интегрирана мултимедийна мрежа, която ще е необходима в близкото бъдеще.

Заради пиковия характер на гласовия/видео/информационния поток, понякога количеството трафик превишава пропускателната способност на връзката. Какво ще се случи в този момент зависи от маршрутизатора. Дали ще буферира трафика в една опашка и ще разреши на първия пристигнал пакет да бъде и първия напуснал през изходния интерфейс, или ще поставя пакетите в различни опашки и ще обслужва определени опашки по-често от другите, зависи от инструментите за управление на претоварванията.

Такива инструменти са формирането на опашки на базата на:

- приоритета (Priority Queuing, PQ);
- потребителя (Custom Queuing, CQ)
- равноправно разпределение на теглото (Weighted Fair Queuing, WFQ)
- класа на трафика и равноправно разпределение на теглото (Class-Based Weighted Fair Queuing, CBWFQ).

Понятието ниво на услугата, се отнася до реалните възможности за налагане на QoS от точка до точка, означаващо способността на мрежата да предоставя услуги, необходими за специфичен мрежов трафик от край до край. Услугите се различават по тяхното ниво на *QoS стриктност*, което показва колко стриктно дадена услуга може да се придържа към специфична честотна лента, закъснение, джитер и загуба на пакети.

Три основни нива на QoS могат да бъдат доставени от край до край през хетерогенна мрежа, както е показано на фиг.7.2:

• **Услуга с най-добро обслужване (Best-effort service)** – Познато като липса на QoS, услугата за най-добро обслужване главно се свързва с липсата на гаранции. На този принцип работят FIFO (First In First Out) опашките, които не правят разлика между различните потоци.

• **Диференцирана услуга (DiffServ), също наричана меко QoS** – При нея един трафик се третира по-добре от останалите, като се осигурява по-бързо обработване, повече средна честотна лента и по-ниско средно ниво на загуби на пакети. Тази услуга предоставя статистическо предпочитание, но не дава твърда гарантираност. Постига се чрез класифициране на трафика и използване на инструменти за QoS като PQ (Priority Queuing), CQ (Custom Queuing), WFQ (Weighted Fair Queuing) и WRED (Weighted Random Early Detection).

• **Гарантирана услуга (Guaranteed service), също наричана твърдо QoS** – Това представлява абсолютна резервация на мрежови ресурси за специфичен трафик. Тази услуга може да бъде постигната чрез протокол за резервиране на ресурси (RSVP - Resource Reservation Protocol) и базирано на

класа на трафика и равноправно разпределение на теглото CBWFQ(Class-Based Weighted Fair Queuing).

За да даде приоритет на определени потоци, първо потокът трябва да се идентифицира и ако е нужно да се маркира. Решаването на тези две задачи, най-често се описва като *класифициране*.

Първоначално идентификацията е започнала да се извършва чрез листи за контрол на достъпа (ACL). Листите за контрол на достъпа идентифицират трафика според инструментите по управление на претоварването като **PQ** и **CQ**. Тъй като **PQ** и **CQ** се настройват в мрежовия възел посредством администриране на маршрутизатора, идентифицирането на пакетите се извършва единствено за този маршрутизатор. Така, настройките за приоритет при QoS важат само за текущия възел и не се предават на следващите маршрутизиращи възли в мрежата. В някои случаи **CBWFQ** класифицирането важи само за настройвания маршрутизатор. Обратен е случая с настройването на битове за IP предимство.

Технологии като *маршрутизиране според политика и гарантирана скорост на достъп* (CAR) могат да бъдат използвани за настройване на предимство, базирайки се на класификация чрез разширени листи за контрол на достъпа. Това позволява значителна гъвкавост за назначаване на предимство, включително назначаване според потребител или приложение, адрес на източника и получателя и т.н.. Обикновено тази функционалност се внедрява колкото се може по-близо до краищата на мрежата или административния домейн, така че всеки последвал мрежов елемент да може да осигури услуга, базирана на определената политика.

Мрежово-базираното разпознаване на приложение (NBAR) се използва за да се идентифицира трафика по-детайлно. Например URL частта от HTTP пакет може да бъде разпозната. След като пакетът се идентифицира, може да бъде маркиран чрез битовете за предимство.

QoS политика чрез базирано на политика маршрутизиране

Базираното на политика маршрутизиране (Policy-Based Routing, PBR) дава възможност да се класифицира трафик посредством критерии, които се описват в разширени листи за контрол на достъпа. Може да се извърши настройване на битове за IP предимство, като е възможно да се направи маршрутизиране към специални трафично-проектирани маршрути, които да позволяват специфично QoS по дължината на мрежата. Като се настройват нивата на предимство за входящия трафик и се използват в комбинация с инструментите за формиране на опашки, е възможно да бъде създадена диференцирана услуга. Тези инструменти осигуряват мощни, прости и гъвкави възможности за внедряване на QoS политики в мрежата.

Използвайки базираното на политика маршрутизиране, могат да се създават карти на маршрутите, които да отговарят на определени критерии за потоците, а след това да се настройват битовете за предимство при съвпадение с изискванията на ACLs.

Динамично идентифициране на потоци

Нов метод за класифициране е мрежово – базираното разпознаване на приложенията (Network Based Application Recognition, NBAR). Всъщност NBAR е просто инструмент за идентифициране, но може да се разгледа като инструмент за класифициране. Както и при всеки един инструмент за класификация, по-трудната част е идентификацията на трафика. Последващото маркиране на пакета е относително лесно, но NBAR прави процеса на идентифициране на друго ниво. Гледайки по-дълбоко във вътрешността на пакета, може да се осъществи идентификация въз основа на URL или MIME типа на HTTP пакета. Тази способност придобива все по-важно значение, тъй като все повече приложения стават web-базирани. В един момент ще бъде оценена нуждата да се различи платечно нареждане от обикновено сърфиране в Интернет. В допълнение към това, NBAR може да идентифицира различни приложения, които използват краткотрайни временни портове. Това се прави като се преглеждат и контролните пакети, за да се определи кои именно портове решава да използва дадено приложение за момента при предаване на своите данни.

NBAR добавя ред предимства, които го правят изключително ценно. Едно от тях е способността да разпознава протоколите. Това позволява на NBAR да следи протоколите действащи на даден интерфейс, да ги идентифицира и да дава статистика за всеки един от тях.

Друга черта на NBAR са модулите за описание на пакетния език (Packet Description Language Module, PDLM), даващи възможност за лесно добавяне на нови разпознавани протоколи към съществуващия списък. Тези модули се създават и се зареждат във Flash паметта, и след това се използват чрез RAM. По този начин се дава възможност за постоянно усъвършенстване на системата без да се налага обновяване или рестартиране на маршрутизатора.

Един от начините по който мрежовите елементи могат да се справят с претоварване на постъпващия трафик е да се използва алгоритъм за формиране на опашки, с помощта на който се сортира трафика. След това се определя метод за приоритизирането му при прехвърлянето му към изходящата линия. Като основни се предлагат следните възможни инструменти за формиране на опашки:

- Първи на входа, първи на изхода (FIFO – First-In, First-Out)
- Приоритетно формиране на опашка (PQ – Priority Queuing)
- Потребителски формирана опашка (CQ – Custom Queuing)
- Базирано на равноправно разпределение на теглото (WFQ – Weighted Fair Queuing)
- Базирано на класа на трафика и равноправно разпределение на теглото (CBWFQ – Class-Based Weighted Fair Queuing).

Алгоритмите за формиране на опашки оказват ефект когато се наблюдава претоварване. По дефиниция, ако линията не е претоварена, няма нужда да се формират опашки от пакети. В условия на липса на претоварване, всички пакети се доставят директно към изходния интерфейс.

Всеки алгоритъм за формиране на опашки е разработен да решава специфичен трафичен проблем и също така има специфичен ефект върху работата на мрежата.

FIFO: Основна възможност за запаметяване и изпращане

В своята най-проста форма (фиг.7.6), FIFO формирането на опашки включва съхраняване на пакетите при претоварена линия и тяхното изпращане в реда на постъпването им, когато линията не е претоварена.

FIFO е много разпространен тип опашка, но си има своите предимства и недостатъци.

Предимства:

- при софтуерно базираните рутери FIFO използва изключително малко от ресурсите на системата в сравнение с по-сложните опашки.

- поведението на FIFO опашките е много предсказуемо – пакетите не се пренареждат и максималното закъснение(delay) се определя от големината на опашката.

- докато размера на опашката се поддържа малък, FIFO осигурява просто разрешение на спора за системните ресурси без да добавя особено закъснение на всеки хоп.

Недостатъци:

- единична FIFO опашка не позволява на рутерите да организират буферирани пакети и да обслужват един клас от общия трафик отделно от останалите класове.

- единична FIFO опашка въздейства върху всички потоци по един и същи начин, заради това закъснението в опашката нараства за всички потоци ако се появи натоварване в мрежата. В резултат на това FIFO може да повиши закъснението, да доведе до поява на цифрово ехо и до загуби в приложение с реално-времеви характер, които преминават през FIFO.

- по време на периоди на натоварване FIFO толерира UDP потоците пред TCP потоците. Когато се загуби пакет поради натоварване, TCP базираните приложения намаляват скоростта с която изпращат данни, докато UDP приложенията не обръщат внимание на загубата на пакети и продължават да изпращат със същата скорост. Тъй като TCP приложенията забавят скоростта на изпращане за да се адаптират към променящите се условия в мрежата, използването на FIFO може да даде резултат в повишено закъснение, цифрово ехо и до намаляване на количеството на изходната ширина на канала използвана от TCP приложенията минващи през FIFO.

PQ. Приоритизиране на трафика

Приоритетното формиране на опашки се грижи за важния трафик. Това му дава възможност да получава най-бързото обслужване във всяка една точка, където е приложено.

Тази технология е проектирана така, че да дава стриктен приоритет за важния трафик. PQ може гъвкаво да приоритизира в зависимост от мрежовия

протокол (IP, IPX, AppleTalk), входящия интерфейс, размера на пакета, адреса на източника или дестинацията и т.н.

При PQ всеки пакет бива поставен в една от четирите формирани опашки – висока, средна, нормална или ниска – според предварително зададен приоритет. Пакетите, които не попадат в класификацията за приоритет се разпределят в нормална опашка (фиг.7.7). По време на предаването към изходния интерфейс, алгоритъмът дава на опашките с по-висок приоритет абсолютно преференциално обслужване за сметка на ниско приоритетните опашки.

PQ е полезно да се прилага, когато е необходимо да се осигури, че трафика с критично значение, преминаващ през различни WAN връзки, получава приоритетно обслужване.

CQ - Гарантиране на честотна лента

Технологията CQ (Custom Queuing) е проектирана за да позволи различни приложения или организации да споделят мрежата между приложения със специфични изисквания към честотната лента или закъснението. В тези среди, наличната честотна лента трябва да бъде споделяна пропорционално между приложения и потребители. Технологията CQ може да се използва за да се осигури гарантирана честотна лента в потенциална точка на претоварване. Тя дава на определен трафик фиксиран дял от наличната честотна лента, като предоставя другия честотен дял за останалия трафик. Формирането на „потребителски“ опашки осигурява трафика, като задава специфично количество от пространството за опашки на всеки клас пакети.

Както и PQ, CQ технологията бива конфигурирана според статистическата информация и не се адаптира автоматично към променящите се мрежови условия.

Поточно – базирано WFQ: Създаване на справедливо разпределение между потоците

За ситуации, в които е желателно да се осигури достатъчно време за реакция, както за леки, така и за тежки мрежови потребители без да се добавя допълнителна честотна лента, решението е поточно – базираното справедливо разпределение при формирането на опашки.

WFQ работи така, че опашките да не изпитат недостиг на честотна лента и трафикът да получава предвидимо обслужване. Малко обемните трафични потоци, които представляват по-голямата част от трафика, получават подобрена услуга, тъй като им се препредават същия брой байтове както голямо-обемните потоци.

Това поведение изглежда като преференциално обслужване на малко-обемния трафик, но всъщност създава справедливост и коректност при обработката на различните потоци (фиг. 7.9).

Технологията WFQ е проектирана да минимизира усилията по конфигуриране и настройката. Тя автоматично се адаптира към променящи се условия на мрежовия трафик. В действителност, WFQ върши добра работа за повечето приложения и затова е избран да бъде механизъм по подразбиране за

формиране на опашките при повечето серийни интерфейси, работещи на скорости по-малки или равни на E1 (2.048 Mb/s).

Базирано на клас WFQ: Осигуряване на мрежов честотен капацитет.

Базираното на клас и тегло равноправно формиране на опашки (CBWFQ) е един нов инструмент за управление на претоварванията, който дава по-голяма гъвкавост. Когато целта е да се осигури минимално количество честотна лента, се използва CBWFQ. В другия случай, когато целта е да се осигури максимално количество честотна лента, се използват *гарантирана(договорена) скорост на достъп* CAR и формиране на трафика.

CBWFQ позволява на мрежовия администратор да задава класове с гарантирана минимална скорост. Тук вместо да се осигурява опашка за всеки индивидуален поток, се дефинира клас, който се състои от един или повече потоци. На всеки клас след това може да бъде гарантирано минимално количество от честотната лента.

Един пример в който CBWFQ може да бъде използвано, е когато трябва да се предотврати потискането на високо-приоритетен поток от множество по-ниско приоритетни потоци.

При CBWFQ може също така да се резервира минимално част от честотната лента. Ако е налична по-голяма част от нея, тогава класът може свободно да я използва. Но идеята е, че това е гарантиран минимум от честотната лента. Също така, ако даден клас не се възползва напълно от гарантираната си минимална скорост, то остатъка може да се използва и от други приложения.

В допълнение, може да се зададе и формиране на ниско-латентни опашки (Low-Latency Queue, LLQ), които всъщност са приоритетни опашки. Тази технология също е позната като приоритетни опашки базирани на класове и тегло със „справедливо“ формиране PQCBWFQ (Priority Queue Class-Based Weighted Fair Queuing).

Механизма LLQ позволява даден клас да бъде обслужван от опашка със стриктен приоритет. Трафикът в този клас ще бъде обслужван преди всеки друг клас. Първо се прави резервиране на определен капацитет от връзката и след това всеки трафик надвишаващ тази резервация се отхвърля.

Извън CBWFQ може да се използва IP RTP приоритет или IP RTP резервиране за да се осигури сходно обслужване само за трафика пренасящ протокола за реално време (RTP – Real Time Protocol).

Избягването на претоварванията е форма на управление на опашките. Техниките за управление на претоварванията, работят за да контролират претрупванията, след като те се осъществят. За разлика от тях, *техниките за предотвратяване на претоварванията* следят натоварването на мрежовия трафик в усилието си да предусетят и да предотвратят очакваното претоварване във възли с натрупване. Основната техника за избягване на претоварванията е теглово ранно произволно откриване (Weighted Random Early Detection, WRED).

WRED: Избягване на натрупванията

Алгоритмите на *произволно ранно откриване* (Random Early Detection, RED) са проектирани за да се избегне претрупванията в мрежите преди те да се окажат въздействие на потоците. RED работи като следи трафичните натоварвания в дадени точки на мрежата и стохастично изключва пакети, когато натрупването започне да се увеличава. Резултатът от това изключване е, че източникът открива отхвърления трафик и забавя скоростта си на предаване. RED е проектиран основно да работи с TCP в IP междумрежовите пространства.

WRED обединява възможностите на RED алгоритмите със технологията IP предимство. Тази комбинация дава възможност за преференциално обслужване на трафика от високо-приоритетните пакети. Тя също така може и избирателно да изключва ниско-приоритетен трафик, когато даден интерфейс започне да изпитва натрупване и може да осигури диференцирани характеристики на работата за различни класове от услуги

Поточно RED

Технологията WRED преди всичко се използва за TCP потоци, които намаляват скоростта на предаване, ако бъде изхвърлен пакет. Съществуват обаче и не свързани с протокола TCP потоци, които нямат механизъм за забавяне при изключване на пакети. В този случай се използва поточното RED.

Тук подходът е да се увеличи вероятността за изключване на даден поток, ако той превиши своя праг.

Поточно-базираното RED разчита на следните два основни подхода за да облекчи проблема с линейната загуба на пакети:

- Класифицира постъпващия трафик в потоци според дадени параметри, като адреси на източника или получателя и техните портове.
- Поддържа информация за състоянията на активните потоци, които са потоците притежаващи пакети в изходящите опашки.

Поточно-базираното RED използва това класифициране и информация за състоянието на потоците за да се увери, че никой поток не консумира повече от позволения му дял от ресурсите на изходния буфер. При този подход се определя кои потоци монополизират ресурсите и започва да ги ограничава.

Поточно-базираното RED осигурява безпристрастност между потоците по следния начин: Следи броя на активните потоци, които съществуват на изхода на даден интерфейс. Според броя на активните потоци и размера на изходящата опашка определя броя на наличните буфери за всеки поток.

7.3 Реализация на QoS

7.3.1 Опашка тип FIFO

Опашките от типа FIFO са най-простият тип опашка, който би могъл да се използва. При нея, всички пакети се третират еднакво, като се поставят в една опашка, и в следствие се обслужват в реда на пристигането им. FIFO опашките може да бъдат срещнати и под името FCFS – First Come, First Served.

В Линукс, както и в повечето рутери по света, FIFO е опашката, която се използва по подразбиране (default). В случай че не бъде указано друго, Линукс

организира своите интерфейси с този тип опашка. Следващата команда показва как чрез `tc` можем да създадем FIFO опашка на Етернет интерфейса `eth0`.

```
#tc qdisc add dev eth0 root pfifo limit 10
```

Програмата `tc` е тази, чрез която се създадем опашката. `qdisc` показва че ще конфигурираме някакъв тип опашка (може да бъде заместено от `class` или `filter`, ако искаме да конфигурираме клас или филтър); `add` указва добавяне на нова опашка; `dev eth0` указва, че опашката се добавя към Етернет интерфейса `eth0`; `root` указва, че това ще е главната опашка (по принцип това не важи за FIFO защото тя е безкласов тип опашка, но се изисква да бъде написано за да се спази синтаксиса на командата); `pfifo` означава, че опашката е от типа `pfifo` (*packet-fifo*); `pfifo` изисква един параметър - `limit 10`, който указва че дължината на опашката (броя пакети, които може да съдържа в себе си) е 10 пакета.

Информация за създадената опашката можем да бъде изведена чрез командата:

```
#tc qdisc show dev eth0
```

В случая се извежда следния резултат:

```
qdisc pfifo 8001: dev eth0 limit 10 p
```

`tc` отговаря като казва, че е налична `pfifo` опашка с номер `8001`: (което по принцип значи `8001:0`) с капацитет 10 пакета. Опашките и компоненти се номерират, или по-скоро се идентифицират чрез 32-битова номерация която е съставена от 16-бита главна част и 16-бита подчинена. Подчинената е винаги 0 за опашките.

Когато се добавя `pfifo` опашка може да не бъде указан номер, с който да се представя тя в последствие. Заради това `tc` задава такъв – `8001:0`.

Изтриването на опашка се извършва със следната команда:

```
#tc qdisc del dev eth0 root
```

Глава осма

Логови файлове. Системи за засичане на нарушения

8.1 Наблюдение на логови файлове

За да бъде подсигурана една система, тя трябва да разполага с изчерпателни, акуратни и внимателно наблюдавани файлове-дневници. Дневниците имат няколко различни предназначения. Първо, те помагат за откриването и отстраняването на практически всякакви проблеми в системата и приложенията ѝ. Второ, те предоставят ценна информация за евентуални злоупотреби със системата. Трето, ако всичко друго пропадне (независимо дали става въпрос за срив или пробив на системата), дневниците могат да дадат важна информация за случилото се.

Правилното конфигуриране на системните механизми за работа с дневници е от изключителна важност за сигурността на системата и цялостната ѝ работа. Но всички положени усилия могат да бъдат провалени, ако формираните в процеса на работа дневници станат твърде големи и запълнят файловата система.

Повечето дистрибуции на Linux освен `syslog` включват и предварително настроена схема за редуване на дневници. Подобно на `syslog`, тази стандартна схема върши добра работа за повечето потребители, но същевременно самата тя представлява твърде важен механизъм, за да може да бъде игнорирана. Необходимо е да бъде разбрано как действат инструментите за управление на дневници, периодично те да бъдат оценявани, както и ако се наложи да бъдат променени настройките на инструментите за управление на дневници.

Всички механизми за управление на дневници включват и някакъв метод, който периодично премества или преименува даден файл-дневник в архивно копие и създава нов (празен) файл. Редуването на дневници е задължително, когато се пазят няколко архивни копия на един файл.

За ежедневно редуване на дневници може да се използва скрипт, наречен `/etc/cron.daily/aaa_base_rotate_logs`. Този скрипт не би трябвало да се променя директно; неговото поведение се определя от файла `/etc/logfiles`, който представлява просто списък с файлове които искате да редувате, заедно с максимално допустимия за тях размер, техните права за достъп и собственици, както и стартиращия скрипт (ако има такъв), който трябва да бъде рестартиран след приключване на редуването.

Препоръчително е да бъде използван отделен дисков дял `/var` на всяка машина, която работи като сървър, защото много по-малко вероятно е един препълнен дисков дял `/var` да предизвика нарушаване на нормалното функциониране на системата, отколкото един препълнен основен дисков дял.

Програмата за системно документиране може да бъде конфигурирана и тествана чрез редица финни настройки с цел съобщенията да се сортират по тип и важност, преди да бъдат записани в дневниците на съответните приложения, както и на централния документиращ сървър. Освен това може да бъде включена и настроена за използване схема за редуване на дневници, която пази

точно толкова стари данни, колкото могат да бъдат използвани според желанието на администратора.

syslog

syslog може да приема данни от ядрото (посредством klogd), от всеки локален процес и дори от процеси на отдалечени системи. Освен това syslog се отличава с гъвкавост, позволяваща да бъде определено какво трябва да се запише в дневниците.

На практика всички варианти на Unix и Linux включват предварително конфигурирана инсталация на syslog като част от базовата си операционна система. Малко системни администратори настройват syslog допълнително да записва нещата, които са важни за техните системи и мрежи, както и да отхвърля ненужните данни, за да не запълват излишно дневниците. Това е сериозен проблем, тъй като натрупването на излишна информация е едно от най-големите предизвикателства в системното администриране.

При зареждане на системата автоматично се изпълнява скрипт, който стартира и общата програма за системно документиране (най-вероятно това е /etc/init.d/syslogd или /etc/init.d/sysklogd в зависимост от използваната дистрибуция на Linux). Чрез този скрипт се стартира и демон klogd, който се явява демонът за документиране на съобщения от ядрото на Linux.

По подразбиране klogd насочва съобщенията от ядрото към програмата за системно документиране и затова повечето хора изобщо не се занимават с него. Но чрез него може да бъде контролирано обработването на съобщенията от ядрото, като се редактира конфигурационния файл на syslogd.

klogd може да бъде извикан като самостоятелна програма за документиране; тоест той може да изпраща съобщенията от ядрото директно към конзола или файл-дневник. Ако не е стартиран като демон, klogd може да се използва за извеждане на съдържанието на буферите на ядрото към файл или към екрана.

Syslog-ng може да изпраща отдалечени съобщения чрез протокола TCP и следователно може да бъде използван заедно със stunnel, ssh и други инструменти за подобряване на сигурността. Тъй като за изпращане на съобщения към отдалечени системи syslog използва единствено еднопосочния протокол UDP, който не може "да се постави в тунел" чрез stunnel или ssh, възможностите за подсигуриране на syslog са далеч по-малки, отколкото за Syslog-ng.

Syslog-ng ("syslog от следващо поколение") представлява опит за увеличаване на гъвкавостта на syslog чрез добавяне на по-добри механизми за филтриране на съобщения, препращане на пакети и евентуално (макар и все още не) на механизми за криптиране и проверка на цялостта на съобщенията. Освен всичко това Syslog-ng поддържа и възможност за отдалечено документиране по протоколите TCP и UDP. Syslog-ng е създаден, разработван и поддържан предимно от Балаз ("Бази") Шайдлер.

Конфигурирането на Syslog-ng изисква далеч повече усилия, отколкото на syslog, но това е признак за неговата гъвкавост.

8.2 Системи за засичане на нарушения

Изчерпателните файлове-дневници, за предпочитане в комбинация с някакъв механизъм за наблюдение и уведомяване, дават своевременна информация за състоянието системната сигурност (освен че представляват безценна помощ при възстановяване на събитията след срив на системата или след някакъв инцидент със сигурността). Но възможностите, които предоставят дневниците, разглеждани като инструмент за подобряване на сигурността, не са безкрайни. Те зависят пряко от генериращите ги процеси в операционната система и приложенията. Събитията, които не се очакват от тези процеси и приложения, могат да бъдат записани в дневниците просто с някакво съобщение в общ вид или, още по-зле, да бъдат пропуснати напълно. Освен това какво ще стане, ако някой промени неправомерно самите процеси, приложения или дневници им?

Тук се намесват Системите за засичане на нарушения (СЗН, Intrusion Detection Systems, IDS).

Една система СЗН (host-based IDS) може да предупреди за извършването на неочаквани промени във важни системни файлове според състоянието на някои съхранявани контролни суми.

Една мрежова СЗН (network IDS) може да предупреди за потенциална атака, основавайки се на база данни, съдържаща известни характеристики и признаци на различните атаки, или дори на различия между текущото състояние на мрежата и състоянието, което СЗН счита за нормално (базова линия, base line).

Но на практика може да бъде постигнато едно добро ниво на защита срещу нарушители, без да се полагат особено големи усилия, чрез използването на безплатни, добре документирани инструменти като Tripwire Open Source и Snort.

Принципи на системите за засичане на нарушения

На практиката съществуват две основни категории СЗН: **системни** и **мрежови**. Системните системи за засичане на нарушения, както подсказва името им, работят само на една система и съответно защитават само нея. От друга страна, мрежовите системи за засичане на нарушения работят на една или няколко машини (всяка от които може да бъде избрана за специален "мрежов наблюдател") и защитават всички системи, свързани към тази мрежа.

Къде трябва да се поставят мрежовите СЗН?

За повечето организации има три основни зони, в които е добре да се поставят "сонди" (т.е. системи, които проверяват мрежовия трафик) - във вътрешната мрежа, в мрежата ДМЗ и от външната страна на защитната стена. Сондата извън защитната стена ще получава най-много фалшиви предупреждения за атаки, но там също така е най-вероятно да се видят неуспешни опити за атаки, сканирания на портове и други типове "рединцидентна" активност.

В ДМЗ по принцип би трябвало да се виждат всички атаки към публично достъпните сървъри, които са успели да преминат през защитната стена, но тук също ще се срещат и много фалшиви атаки.

Във вътрешната мрежа не би трябвало да виждате такива.

Все пак трябва да се реагира незабавно на всички (реални) атаки, които са успели да стигнат до вътрешната мрежа.

Във всеки случай включената сонда ще може да вижда нещо само ако:

- Локалната мрежа, към която е свързана, използва комутатор с дублиращ порт.
- Локалната мрежа използва концентратор или друг тип споделена среда.

Може да бъде вмъкнат концентратор и да се включи сондата директно в мрежата в критична възлова точка - например непосредствено между защитната стена и вътрешната мрежа.

По този начин няма да се прихванат атаките между вътрешните системи, но ще могат да бъдат засичани атаки от или към Интернет.

Сондата трябва да се разположи на физически сигурно място, особено в случая с последната от горните точки.

Системни СЗН: програми за проверка на цялостта

Обикновено системните СЗН разчитат предимно на механизми за проверка на цялостта. На теория те би трябвало да използват много по-широк набор от инструменти. Комерсиалните продукти за засичане на нарушения като ISS RealSecure и Network Filght Recorder на Маркъс Ранум при нужда могат да работят с по-усъвършенствани методи (като анализиране на трафика) и само на една отделна машина.

Проверките на цялостта включват създаването и поддържането на защитена база данни с контролни суми, криптографски хешове и други характеристики на жизненоважните системни файлове на дадената машина (и на всичко останало, което не се очаква да бъде променяно).

Програмата за проверка на цялостта периодично сравнява текущото състояние на тези файлове с информацията в базата данни: ако някой файл бъде променен, се генерира предупреждение или съобщение за грешка. В идеалния случай тази база данни би трябвало да бъде съхранявана на файлова система, достъпна единствено за четене, или даже отделно от машината, за да се предотвратят опитите за неправомерното ѝ променяне.

Като се проверяват редовно системните програми и други важни файлове с помощта на базата данни, може да бъде намалена вероятността системата да бъде пробита, без изобщо да се разбере за реализирания пробив в сигурността. Колкото по-бързо администраторите научат за даден пробив в системата, толкова по-големи са шансовете те да успеят да заловят или поне да изгонят нарушителите, преди да нанесат твърде много щети.

Очевиден е факта, че не е задължително да се знае как точно е бил променен някой от наблюдаваните файлове (въпреки че и това би било добре да се знае). По важното е да се знае най-вече дали той е бил променян.

Трябва да се отбележи, че една добра програма за проверка на цялостта ще посочи променените външни характеристики на /bin/ls: размер, дата на променяне, физическо разположение (индексен описател) и т.н.

Една програма за проверка на цялостта с ненадеждна база данни е напълно безполезна. Изключително важно е да бъде създадена база данни възможно най-скоро след инсталиране на операционната система от надежден източник. С други думи инсталирането, настройването и поддържането на една програма за проверка на цялостта не си струва усилията, ако нейната база данни не е била създадена на чиста система.

Друго нещо, което трябва да се имат предвид за програмите за проверка на цялостта, е, че те не са превантивни (освен ако една или повече от системите в граничната мрежа не "поеме удара", която нарочно е оставена да бъде атакувана, за да може да се разбере навреме за атаката и да се защитят останалите системи. Само че нищо не гарантира, че нападателите ще атакуват именно тази система!) В повечето случаи, когато програмата за проверка на цялостта открие нередност, се разполага със съвсем малко време за намеса и предотвратяване на атаката. Освен това нападателят може да прихване е да заглуши предупреждението, преди то да успее да информира администратора.

Това не означава, че проверките на цялостта са напълно безсмислени! Напротив, първата стъпка при реагирането на даден инцидент е да се разбере, че въобще нещо се е случило.

В случай, че на системата има инсталирана програма за проверка на цялостта, много по-бързо ще бъде разбрано за започната вече атака и да се предприемат съответните ответни действия. Дори ако се случи най-лошото, данните от програмата за проверка на цялостта ще бъдат изключително полезни, когато администратора се опитва да разбере какво точно се е случило и когато се започва самото възстановяване на системата след атаката.

В случай, че администраторът желае да направи всичко възможно, за да засече атаките, преди те да достигнат дадена система, ще е необходимо да бъде използвано нещо по-усъвършенствано - т.е. нещо в добавка към системите за проверка на цялостта.

Мрежови СЗН: търсене на признаци и аномалии

Системните системи за засичане на нарушения обикновено са само от един тип (програми за проверка на цялостта), но мрежовите СЗН биват два основни вида: такива, които разчитат на признаци на атаките (характеристики на мрежовия трафик, специфични за определени атаки), и такива, които са достатъчно интелигентни, за да могат да засичат потенциални атаки в зависимост от отклоненията спрямо нормалното състояние на мрежата. Най-често използваните МСЗН разчитат най-вече на сканиране за признаци, като много от тях притежат и някаква функционалност за засичане на аномалии.

Освен системите за откриване на признаци и за засичане на аномалии има и други видове мрежови СЗН. Повечето от тях попадат в категорията, която носи името "анализиращи". Тези системи записват колкото е възможно повече данни, но ги анализират едва след протичането на въпросните събития. В известен смисъл това е един изключително мощен подход, защото включва

способност за запаметяване на специфичните признаци дори на най-потайните и сложни атаки.

Проблемът е, че анализиращите СЗН могат да се ползват при пълните им възможности, едва след като системата е изпитала множество завършени атаки, когато в повечето случаи е твърде късно.

Системи за откриване на признаци

Системите за откриване на признаци са най-разпространеният вид мрежови СЗН по няколко причини.

Първо, те са най-простите: характеристиките на мрежовите транзакции се сравняват с известните признаци на атаките и ако дадена транзакция наподобява някоя известна атака, системата ще запише предупреждение в дневниците (а вероятно ще го изпрати и на нечий пейджър).

Второ, те са лесни за поддържане: всичко, което трябва да се направи, е да се обновяват редовно базата данни с признаците.

Трето, процентът на генерираните фалшиви тревоги е относително малък, което се цени изключително много от системните администратори, които и без това получават твърде много е-поща, свързана с мрежови проблеми.

Системите за откриване на признаци, които се наричат още "откриватели на злоупотреби", представляват успешен и практичен подход за засичане на нарушения. Те обаче имат едно важно ограничение: тъй като разчитат на признаците на известните досега атаки, тези системи не са особено ефективни срещу нови атаки и варианти на стари атаки, чиито признаци се различават достатъчно от оригиналните. Трябва да се има предвид, че признаците на повечето атаки са станали известни едва след като някой е станал тяхна жертва.

Системи за засичане на аномалии

Системите за засичане на аномалии, наричани понякога системи за проверка на състояние, се използват далеч по-рядко.

Първо, те обикновено са доста сложни: определянето на елементите, съставлящи "нормалния" трафик за дадена мрежа, не е тривиална задача за човешкия ум, ето защо всяка автоматизирана система, която иска да прави това, ще трябва да притежава много добър изкуствен интелект.

Второ, тези системи се поддържат трудно: дори когато разполагат с добър изкуствен интелект и отлични механизми за статистическо моделиране, системите за засичане на аномалии обикновено се нуждаят от продължителен и понякога труден "инициализационен" период, по време на който трябва да съберат достатъчно мрежови данни, за да създадат статистически достоверен профил на нормалните състояния на мрежата. След това системата изисква често, периодично (до безрайност) допълнително настройване.

Трето, дори и след като цялата тази работа бъде извършена, системите за засичане на аномалии обикновено генерират много повече фалшиви тревоги от системите за откриване на признаци (макар, че с времето този проблем намалява). Това може да причини доста неудобства.

Snort

Програмите за проверка на цялостта могат да служат като аларми срещу нарушителите. Но като такива те са по-полезни след атаката, а не по време на нея: обикновено когато нападателите започнат да променят файлове на дадена система, атаката им вече е успяла. Това се дължи на факта, че програмите за проверка на цялостта са ограничени до локалната система: те работят с локални файлове, а не с мрежови пакети. За по-активно засичане на нарушения (засичане на опити за нападения или на атаки в момента на протичането им) ще се наложи да бъдат наблюдавани опитите за нападения и текущите атаки, докато все още са в мрежата, но преди да достигнат до нашите системи.

Безспорният шампион при мрежовите системи за засичане на нарушения с отворен код е ***Snort***. Snort е великолепно и гъвкаво творение.

Първо, като подслушвач на пакети (или ако предпочитате по-официалния термин "анализатор на протоколи") Snort превъзхожда tcpdump. Като подслушвач на пакети Snort е изключително бърз, изчерпателен и удобен за ползване (поне от гледна точка на компютърните манияци).

Второ, Snort може да следи пакетите. Snort може да запазва цели последователности от мрежовия трафик, чрез които да проследявате и ясно да посочвате нарушителите.

Трето, Snort е 100% конфигурируема мрежова система за засичане на нарушения, разполагаща както с библиотека с белези на атаки ("правила"), така и с механизъм за задаване на правила от потребителя. Snort не само се равнява на, но в някои случаи дори е по-добър и по-бърз от скъпите комерсиални системи за засичане на нарушения. В това отношение Snort е GIMP, Apache и Nessus на системите за засичане на нарушения.

За разлика от някои комерсиални СЗН, Snort позволява да се съставят свои собствени правила и дори цели системи за проверка ("приставки на Snort").

Това е една важна функционалност, защото непрекъснато се появяват нови атаки.

Тъй като всяка нова версия на Snort е още по-усъвършенствана от предишната и следователно по-ефективна при засичане на подозрителна мрежова активност, е препоръчително да се сваля и компилира изходния код на последната версия на Snort, или да се използва предварително компилирания пакет, предоставен от разработчиците на Snort.

Глава девета

Управление на мрежи

9.1 Управление на мрежи

SNMP (Simple Network Management Protocol) протокол на приложно ниво, който не е ориентиран към обслужване на крайните потребители, а предлага функционалност, свързана с управление на комуникационните протоколи в крайните компютри (host) и активното мрежово оборудване (мостове, комутатори, маршрутизатори, шлюзове и др.)

Стандартизиращият подхода за управление на мрежовата среда е всички нейни елементи – протоколи, мостове, маршрутизатори, шлюзове да бъдат управлявани като се мрежови обекти. Управлението на един мрежов обект е възможно да се разглежда като в няколко основни аспекта:

- ❖ Управление на отказите (fault management);
- ❖ Управление на производителността (performance management);
- ❖ Управление на протоколите (layer management);
- ❖ Управление на сигурността (security management) и др.

SNMP е приложен протокол, който на базата на стандартната комуникационна платформа трябва да осигури преноса на статусна и управляваща информация, съвместявайки го с транспорта на приложно ориентираните информационни потоци. За да постигне тази функционалност SNMP използва протоколен стек TCP/IP

SNMP осигурява на управляващия процес (manager process) от управляващата станция да обменя ориентирани към управлението съобщения с управляващите процеси, стартирани в архитектурата на мрежовите обекти: крайни машини (host), маршрутизатори, шлюзове и т.н.

Информацията, свързана с управлението на група мрежови обекти, асоциирани в крайна като мащаб мрежова конфигурация се натрупва в управляващата станция като база от данни за управление - MIB (Management Information Base)

Протоколните примитиви на SNMP се пренасят от транспортния протокол UDP и са структурирани в синтактично съгласно ASN.1.

Спецификата на всеки мрежов обект определя общия и специализирания набор от примитиви за достъп до управлението на този обект.

SNMP създава среда за изграждане на множество приложения, функционалността на които се основава на информацията натрупвана и актуализирана в MIB.

За целите на управлението на мрежови обекти в ISO-стека е дефиниран протокола CMIP (Common Management Information Protocol), но приложението му е ограничено.

9.1.2 Управленска информационна база (MIB)

MIB дефинира обекти, които могат да бъдат управлявани във всеки слой на TCP/IP протокола. Има две версии: MIB-I и MIB-II. MIB-I е дефинирана в

RFC1156 и към настоящия момент е класифицирана като *исторически* протокол със състояние “не се препоръчва”.

Всеки управляван възел (node) поддържа само подходящи за него групи. Ако няма портал (gateway), не е необходимо да се поддържа групата EGP. Но ако групата е необходима, то всички обекти в нея трябва да се поддържат.

Списъкът от дефинираните управлявани обекти е изведен от тези елементи, които се смятат за съществени. Този подход не е ограничителен, тъй като SMI дава механизъм за разширяване, например за дефиниране на нова версия на MIB и дефиниране на частни или нестандартизирани обекти.

Таблица 9.1 Дефиниция на групите в MIB

Група	Обекти за	Брой обекти в групата
System	Основна информация за системата	7
Interfaces	Мрежата	23
AT	Преобразуване на адреси	3
IP	Интернет протокол	38
ICMP	Интернет протокол за контрол на съобщенията	26
TCP	Протокол за контрол на преноса	19
UDP	Протокол за потребителски дейтаграми	7
EGP	Протокол за външни портали	18
SNMP	приложни групи в SNMP	30

System group:

- sysDescr - пълно описание на системата (версия, HW, OS);
- sysObjectID - идентификация на обект на Производител;
- sysUpTime - време от последната инициализация;
- sysContact - име на човек за връзка;
- sysServices - предлагани услуги от устройството;
- Interfaces group:
- ifIndex - номер на интерфейс;
- ifDescr - описание на интерфейса;
- ifType - тип на интерфейса;
- ifMtu - размер на най-голямата дейтаграма;
- ifAdminisStatus - състояние на интерфейса;
- ifLastChange - време откакто интерфейсът е влязъл в последното състояние;
- ifNErrors - брой на насочените навътре пакети, съдържащи грешки;
- ifOutDiscards - брой на насочените навън пакети, които са изхвърлени;

Address Translation Table group:

- atTable - таблица за преобразуване на адреси;
- atEntry - всеки ред дава съответствие между един мрежов адрес и един физически адрес;
- atPhysAddress - физически адрес, зависещ от устройството;
- atNetAddress - мрежов адрес, съответстващ на дадения atPhysAddress;

IP group:

- ipForwarding - указва дали тази единица е IP портал;
- ipInHdrErrors - брой входящи дейтаграми, изхвърлени поради грешки в техните заглавни части;
- ipInAddrErrors - брой входящи дейтаграми, изхвърлени поради грешки в техните IP адреси;
- ipInUnknownProtos - брой входящи дейтаграми, изхвърлени поради непознат или неподдържан протокол;
- ipReasmOKs - брой успешно реасемблирани дейтаграми;

ICMP group:

- icmpInMsgs - брой получени ICMP съобщения;
- icmpInDestUnreachs - брой получени ICMP съобщения “недостъпна дестинация”;
- icmpInTimeExds - брой получени ICMP съобщения за изтекло време;
- icmpInSrcQuenchs - брой получени ICMP съобщения “source quench”;
- icmpOutErrors - брой ICMP съобщения, неизпратени поради проблеми в ICMP;

TCP group:

- tcpRtoAlgorithm - алгоритъм за определяне на времеизчакване за повторно излъчване на непотвърдени октети;
- tcpMaxConn - ограничение за броя едновременни TCP връзки, които устройството може да поддържа;
- tcpActiveOpens - брой пъти TCP връзки са минавали в състояние SYN-SENT от състояние CLOSED;
- tcpInSegs - брой получени сегменти, включително тези с грешки;
- tcpConnRemAddress - отдалеченият IP адрес за тази TCP връзка;
- tcpInErrs - брой сегменти, изхвърлени поради грешка във формата им;
- tcpOutRsts - брой генерирани reset-сигнали;

UDP group:

- udpInDatagrams - брой UDP дейтаграми, доставени на потребители;

- udpNoPorts - брой получени UDP дейтаграми, за които липсва приложение на указания порт;
- udpInErrors - брой получени UDP дейтаграми, които не са доставени по други причини, освен липсата на приложение;
- udpOutDatagrams - брой UDP дейтаграми, изпратени от това устройство;

EGP group:

- egpInMsgs - брой получени EGP съобщения без грешки;
- egpInErrors - брой получени EGP съобщения с грешки;
- egpOutMsgs - брой EGP съобщения, генерирани в това устройство;
- egpNeighAddr - IP адрес на EGP-съседа на това устройство.

Това не е пълният списък за MIB, а е даден като пример за обекти, дефинирани във всяка група. Към настоящия момент тези модули поддържат Ipv4.

9.1.3 Опростен протокол за управление на мрежи (SNMP-Simple Network Managing Protocol)

Протоколът SNMP добавя многогодишните подобрения и опит в използването на SGMP и позволи работа с обекти, дефинирани в MIB чрез представяне, определено в SIM.

RFC 1157 дефинира Станция за управление на мрежата (Network Management Station - NMS), от която се стартират приложни програми за управление на мрежата (Network management applications - NMA), с които се следят и контролират мрежовите елементи (network elements - NE), като хостове, портали и терминални сървъри. Тези мрежови елементи използват управленски агент (Management agent - MA) за изпълнение на функции по управлението, поискани от станциите за управление на мрежата. SNMP се използва за обмен на управленска информация между станциите за управление на мрежата и агентите на мрежовите елементи.

Протоколът за проверка (authentication protocol) дава механизъм, чрез който управляващите съобщения, излъчени от даден отдел, могат да бъдат надеждно проверени, че идват именно от този отдел.

Протокол за защита (privacy protocol) дава механизъм, чрез който генерираните от SNMPv2 управляващи съобщения се защитават от отваряне. Принципните опасности, от които предпазва протоколът за защита, са следните:

- модификация на информацията;
- маскиране;
- модификация на потока на съобщенията;
- отваряне.

Следните услуги осигуряват защитата срещу тези опасности:

- цялост на данните - дава се от алгоритъма за обработка на съобщения MD5 (message digest). Върху специална част от SNMPv2

съобщение се пресмята 128 битова стойност (наречена digest), която влиза като част от съобщението до получателя;

- проверка за произхода на данните - осъществява се чрез добавяне на префикс към всяко съобщение, в който има секретна стойност, еднаква за подателя и получателя преди обработката по горния алгоритъм
- закъснение или повторение на съобщение - осъществява се посредством времето на създаване на съобщението, записано в него;
- конфиденциалност на данните - осъществява се чрез симетричния протокол за защита (privacy protocol), който шифрира подходяща част от съобщението чрез таен ключ, познат само на подателя и получателя. Този протокол се използва заедно със симетричния алгоритъм за шифриране, в режим на блоково верижно шифриране, което е част от стандарта за шифриране на данни (Data Encryption Standard - DES). Определената част от SNMPv2 съобщението се шифрова и включва към съобщението до получателя.

Целта на административния модел на SNMPv2 е да дефинира как административната рамка се прилага за реализацията на ефективно управление на мрежите в най-разнообразни среди и конфигурации.

Моделът обхваща използването на ясни идентификации за самостоятелни единици (peer), обменящи SNMPv2 съобщения. Т.е., това е отдалечаване от административния модел, базиран на общност, при първоначалния SNMPv1. Чрез недвусмисленото идентифициране на източника вместо получателя на всяко SNMPv2 съобщение, новата стратегия подобрява историческата схема на общността, както чрез поддръжка на по-удобен модел за контрол на достъпа, така и чрез ефективното използване на асиметрични протоколи за защита (публичен ключ)

Тъй като SNMP има модулна структура, необходимостта от промяна на даден модул в повечето случаи не се отразява пряко на другите модули. Това дава възможност лесно да се дефинира версия3 върху версия 2. Например, за да се добави нов формат на SNMP съобщенията ще бъде достатъчно да се обнови модула за обработка на съобщенията. Освен това, поради необходимостта от поддръжка на версии 1 и 2, възможно е да се добави новия модул в подсистемата за обработка на съобщенията.

9.2 Наблюдение , управление и отстраняване на неизправности в мрежата

На този етап става ясно, че компютърните мрежи — макар и прости като концепция — могат да бъдат доста сложни при изпълнение. Много неща още в самото начало при изграждане на мрежата могат да тръгнат неправилно и администраторът трябва да бъде подготвен за да открие, диагностицира и отстрани проблеми с връзките.

Трябва редовно да се наблюдава мрежата с цел за да се идентифицират и коригират потенциални проблеми. В корпоративна работна среда не е доста-

тъчно само да свържат компютрите един с друг. Времето е важен фактор в днешния бизнес свят и от особена важност е оптимизирането на мрежата за постигане на най-висока възможна производителност. Достъпни са различни инструменти, които могат да помогнат на администратора да наблюдава, управлява и отстранява неизправности в LAN и WAN среди. Те варират от прости TCP/IP помощни програми до интелигентни софтуерни пакети и хардуерни устройства на независими производители.

Анализиране и оптимизиране на производителността на мрежата

Администрирането на компютърна мрежа често е тежка работа, свързана с високо натоварване. Това е в сила не само за мрежа на голяма организация, но и за локална мрежа на малка компания.

Тъй като мрежовият персонал не бездейства дълго, много администратори откриват, че действат в *реактивен (противодействащ)* режим; т.е. те са толкова заети за отстраняване на проблеми след тяхното възникване, че не могат да отделят време за реализиране на мерки, които ще възпрепятстват повторното възникване на тези проблеми.

Макар че допускаме, че вие (като администратор) може от време на време да действате в реактивен режим, той все пак е неефективен и уморителен в дългосрочен план. Поради недостатъците на реактивния режим, трябва да се отдели време, за да бъде изготвен *проактивен* план за управление на мрежата. Този план позволява да бъдат открити малки проблеми, преди те да прераснат в големи такива. Чрез предвиждане на потенциални източници на проблеми и предприемане на мерки за тяхното отстраняване, ще бъде спестено голямо количество време и усложнения. Също така в много случаи ще се спестят и пари на организацията.

Ключовите концепции при анализиране и оптимизиране на производителността на мрежата са следните:

- Тесни места (bottlenecks)
- Базови линии (Baselines)
- Оптимални практики

Тесни места

Тящото място (bottleneck) е точката в системата, ограничаваща *производителността*, която представлява количеството данни, които могат да текат по мрежата. Тящото място в мрежата ограничава данните точно както стеснената горна част на бутилката ограничава количеството течност, която може да се налее или излее от дадена бутилка.

Базови линии

Първата стъпка при определяне на начина за ефективна работа на мрежата включва сравняване на различни измервания (например броя на байтовете, предавани за секунда, или броя на изпуснатите пакети) със същите измервания, направени по-рано.

Базовата линия трябва да бъде измерена в такъв момент, когато мрежата работи нормално. Не трябва да се снемат измервания в най-натовареното

време от деня и не трябва да се чака, докато всички си отидат вкъщи вечерта, за да бъде направено измерването, когато мрежата не се използва.

За да се установи базовата линия, за предпочитане е да се измери производителността на мрежата по време на типичното ѝ използване. Добър начин за това е да се направят няколко отделни измервания на равни интервали и след това да се вземат тяхната средна стойност.

Освен че базовата линия помага да бъдат определени в процеса на разработка тесните места, тя подпомага и в следното:

- Да се идентифицират т.нар. тежки потребители, т.е такива които след включването си товарят мрежата
- Да се съпоставят дневни, седмични или месечни модели на утилизация на мрежата
- Да се балансира цената на ъпгрейд на мрежови компоненти

В много случаи може да се открие, че група потребители на мрежата „изяждат“ голям процент от пропускателната ѝ способност. Идентифицирането на тези тежки потребители дава на администратора благоприятната възможност да направи следното:

- Да ги посъветва относно начините за спестяване на пропускателна способност
- Да ограничи тяхната работа чрез софтуерен контрол
- Да планира тяхната тежка работа и да намери начини, за да предотврати влиянието на тяхната работа върху ефикасността на мрежата

Най-добрите практики представляват препоръчителни начини за извършване на административни и други мрежови задачи по най-ефикасен и ценово ефективен начин. Например разпространителите на продукти за наблюдение на мрежи препоръчват да се записват прихванати пакети във файл, когато се наблюдава мрежата с цел установяване на базова линия.

Това позволява на администратора по-късно той да ги анализира или да ги изпрати на професионални анализатори на мрежи. Също така ще се получи постоянно документиране на извършените наблюдения.

Програмите за наблюдение на мрежа често се означават като *протоколни анализатори*. Повечето анализатори са софтуерно-базирани и позволяват да се прихващат отделни пакети (наричани също кадри или *фреймове*), докато пътуват по мрежата.

По принцип трябва да бъдат прихванати само толкова статистически данни, колкото реално са необходими за оценка на производителността на мрежата, и да се изпълни софтуера за наблюдение (мониторинг) на мрежата по време на периоди с ниска степен на използване, защото самият софтуер за мониторинг влияе върху производителността на системата.

Инструменти за мониторинг и мениджмънт на мрежата

Съществуват много софтуерни пакети, които могат да помогнат на администратора при наблюдението и управлението на мрежата. Някои са включени в мрежовата операционна система, някои могат да бъдат свалени от Web като свободни или споделени, а някои са скъпи и с богати възможности.

Софтуер за мониторинг на мрежи

Софтуерът за мониторинг (наблюдение) на мрежи варира от прост до сложен и от безплатен до скъп.

Интелигентните средства за мрежов мониторинг се наричат протоколни анализатори. Протоколните анализатори прихващат пакети (т.е. фреймове), които се предават между два или повече компютъра или мрежови устройства. След това анализаторът декодира (интерпретира) пакетите така, че да може да се видят данните на английски (или на друг език), съпоставени с двоичния език.

Интелигентният протоколен анализатор осигурява също статистически данни и информация тенденциите за прихванатия трафик.

Терминът *снифър* често се използва за означаване на всяка програма, която позволява да „подслушвате“ мрежовия трафик.

Network Associates регистрира търговската марка Sniffer (и нейната разширена версия Sniffer Pro). И двата продукта представляват мрежови анализатори.

Подслушващите програми имат могат да бъдат използвани от хакери и кракери за извличане на потребителски имена и пароли, които се изпращат по мрежата в обикновен текст. Тези пълномощия след това се използват за придобиване на неоторизиран достъп до системи.

Въпреки това снифърите имат много законни приложения за мрежови администратори, в това число:

- Анализ на проблеми на свързването
- Анализ на производителността
- Откриване на нарушения

Microsoft Performance Monitor и Microsoft System Monitor

Performance Monitor на Windows NT 4.0 (наречен System Monitor в Windows 2000) измерва производителността на голям брой системни компоненти, включвайки броячи на мрежови компоненти.

Мониторът позволява да се показват стойности във форма на графика, да се записват данни в дневник и да се съставя отчет. Измерванията могат да бъдат наблюдавани в реално време, да бъдат обновявания автоматично или да бъдат обновявани при поискване.

Performance Monitor и System Monitor позволяват също така да се конфигурират предупреждения (alerts), които наблюдават зададени броячи и да уведомяват, когато стойността се покачи над или спадне под предварително дефинирана граница.

За да се идентифицирате тесните места в мрежата, трябва да бъдат наблюдавани броячи на мрежовия интерфейс, между които:

- Общия брой байтове за секунда
- Изпратени байтове за секунда
- Приети байтове за секунда

Можете също така да се наблюдават и следните броячи на обекти в протоколния слой:

- Приети сегменти за секунда
- Изпратени сегменти за секунда

- Приети фреймове за секунда
- Изпратени фреймове за секунда

Наблюдението на тези броячи позволява на администратора да планира разпределението на капацитета на пропускателната способност. Например, ако общия брой байтове, предавани в секунда, е близък или равен на максималния капацитет на преносната среда (media) на мрежата, трябва да се помисли или за ъпгрейд на оборудването или за намаляване на натоварването на мрежата.

Снифър технологии

Популярният мрежов анализатор Sniffer прерасна в сложен комплект от мрежови инструменти, в това число:

- Sniffer Pro LAN и Sniffer Pro WAN
- Sniffer Pro High-Speed
- Gigabit Sniffer Pro
- Sniffer Distributed Analysis Suite

Продуктите Sniffer разрешават интелигентно филтриране, базирано на съвпадения в моделите, IP/IPX или DLC адреси.

Sniffer Pro включва генератор на трафик, спомагащ тестването на нови устройства или приложения. Така той може да бъде използван за да се симулира мрежов трафик и да се измерят времената за отговор и броя на скоковете (hops).

Sniffer включва вградени TCP/IP помощни програми, като ping, tracet, DNS lookup и др.

Дисплеят на Sniffer може да бъде конфигуриран да показва информация за пакетите и разпределението по протоколи (визуализирани под формата на диаграма).

Sniffer включва Expert Analyzer, който подпомага и при диагностициране на мрежови проблеми.

Софтуер за мениджмънт на мрежи

Трябва да се разбере разликата между софтуера за мрежов мониторинг и софтуера за мрежов мениджмънт.

Последният по принцип е по-изчерпателен и макар че включва компоненти за мониторинг, той позволява на администратора да прави и много повече.

Управлението на мрежовите услуги представлява голяма част от работата на всеки мрежов администратор. Това е истина особено в работна среда в голяма организация.

Трябва да се познават добре инструментите, които могат да направят тази задача по-лесна, включително възможности за управление, вградени в модерните мрежови операционни системи и софтуерни продукти, предлагани от производители на операционни системи и независими производители.

Управлението на мрежата включва следните задачи:

- Документиране на устройствата в мрежата и състоянието на всяко от тях
- Създаване на инвентарен списък на мрежовия софтуер, който

позволява да се инсталира софтуер и да се извършват обновявания по мрежата

- Измерване на софтуера за осигуряване на данни за използваните приложения и как, кога и от кога са използвани
- Управление на софтуерните лицензи
- Отдалечено управление на клиентски машини и сървъри по мрежата и управление на отдалечени десктопи
- Уведомяване на администраторите за събития като отказ на мрежови компоненти или предварително дефиниран капацитет, който е достигнат или надхвърлен

Софтуер за мениджмънт на малки и средни мрежи

Наред с продуктите за мрежов мениджмънт, предлагани от големи софтуерни компании, като Microsoft, Novell, IBM и Hewlett Packard, многобройни по-малки фирми произвеждат допълнителни продукти, предназначени за пазара на малки до средни мрежи. Това са Network Monitoring Suite (NMS) на Lanware, който използва Simple Networking Management Protocol (SNMP) и осигурява такива възможности, като рестартиране на сървъри, изготвяне график за изпълнение на събития и презареждане на сървъри. ViewLAN на NuLink е друг относително прост SNMP инструмент за мениджмънт и мониторинг.

Устройства за мониторинг на хардуера и отстраняване на проблеми

Повечето хардуерни устройства, които се използват за наблюдение и управление на мрежата, са просто специално предназначени за целта компютри (често преносими), които изпълняват собствен софтуер за протоколен анализ и управление на мрежата.

Отстраняване на проблеми в мрежите

Често администраторът се сблъсква с проблеми по мрежата.

Те варират от постепенни забавяния, които дразнят потребителите, до пълно изгубване на връзка по цялата мрежа, като това прекъсва работата на хиляди служители.

Всички продукти, които бяха разгледани, могат да бъдат използвани за отстраняване на проблеми в мрежата.

Тъй като повечето съвременни мрежи работят с TCP/IP, има на разположение няколко полезни помощни програми за отстраняване на проблеми. Те могат да бъдат използвани без необходимостта от закупуването, инсталирането и изучаването, свързани с един сложен и скъп продукт за мрежов мениджмънт.

Най-основният мрежов проблем е невъзможността на един компютър да комуникира с друг компютър. За изгубването на връзката може да има много причини и те могат да бъдат свързани както с хардуера, така и със софтуера.

Първото правило при отстраняване на проблеми е да се провери физическата връзка.

Преди да се започнат сложни действия по отстраняване на проблеми, трябва да се проверят дали кабелите са правилно включени и в двата края, че мрежовият адаптер функционира (да се провери светлинната индикация на

връзката на самата мрежова карта), че светят светлините за състоянието на хъба и че комуникационният проблем не е просто повреда на хардуера.

Друга често срещана причина за проблеми на мрежата е меко казано „грешката на оператор“. Вероятно причината дадена работна станция да не вижда останалата част от мрежата е, че потребителят се е логнал на локалната машина, но не и в мрежата.

Потребителите трябва да използват правилното потребителско име и парола. Трябва да се провери дали техните акаунти не са ограничени по начин, който им забранява да установяват логическа връзка с мрежата.

Например броят на логванията може да бъде ограничен само до работните часове през дните от седмицата, или потребителят може да бъде ограничен да се логва само от конкретна работна станция.

Хардуерните проблеми са сравнително прости за отстраняване, след като бъдат открити.

Софтуерните проблеми могат да бъдат по-трудни за проследяване. Най-честият виновник е погрешното конфигуриране на софтуера. Настройките на софтуера могат да бъдат променени от инсталационната програма на наскоро инсталиран софтуер, или потребителят може да е експериментирал с настройките. Липсващи или повредени файлове могат да причинят всевъзможни проблеми, включително проблеми с възможността за връзка с мрежата. Потребителите неволно (или по друг начин) изтриват файлове, а пикове в захранващото напрежение или внезапно изключване на компютъра могат да повредят данни във важни файлове. Не трябва да се пренебрегва също и възможността за наличие на вирус.

Каквато и да е предполагаемата причина за проблема, следването на определен набор от предварително планирани стъпки за всеки сценарий на отстраняване на проблеми ще гарантира, че се обхващат всички основни пунктове.

TCP/IP инструментите, които позволяват да бъде тествана връзката до друга машина и да се определи пътя, който изминава пакетът, за да достигне до местоназначението, са следните:

- ping и pathping
- Помощни програми за трасиране

ping е съкращение на *packet internetwork grooper*.

Тази команда е проста помощна програма, която изпраща съобщение, наречено Echo Request, използвайки протокола Internet Control Message Protocol (ICMP), до зададен компютър местоназначение. Компютърът местоназначение отговаря с изпращане на ICMP echo.

Първата стъпка при проверка за предполагаем проблем на връзката е ping-ването на хоста. Ако трябва да се провери връзката към Интернет, трябва да се реализира ping към надежден хост в Интернет, например www.yahoo.com.

Ако се получи отговор, това ще означава, че физическата връзка между двата компютъра е незасегната и работи.

Командата ping може да бъде подадена с използване или на IP адрес, или на името на компютъра местоназначение. За тестване на връзката използвайте IP адреса.

Ако ping по IP адрес се реализира, но не се получава отговор, когато се ping-ва същия компютър по име, това означава, че има проблем със сървъра или конфигурацията за преобразуване на имена.

Такъв проблем може да възникне, когато компютъра не използва DHCP и няма адрес на DNS сървър, въведен в свойствата на TCP/IP. От друга страна, когато компютъра е DHCP клиент, DHCP сървърът може да зададе адреса на DNS сървъра.

Терминът ping time се отнася до количеството време, което изминава между изпращането на заявка за ехо (Echo Request) и приемането на ехо отговор (Echo Replay). Късото време на ping-ване показва бърза връзка.

Командата ping може да бъде използвана също за тестване дали TCP/IP стекът на компютъра е инсталиран правилно и функционира.

Тестът се изпълнява чрез ping до адреса за обратна връзка 127.0.0.1.

Ако се получи отговор, това означава, че стекът работи.

pathping комбинира възможностите на ping с тези на tracert и осигурява допълнителна информация, която иначе не се визуализира от тези програми. С pathping може да бъдат открити кои маршрутизатори предизвикват проблеми по мрежата и да се измери колко пакета се губят при конкретен маршрутизатор.

Помощните програми за трасиране се използват за откриване на маршрута на движение на даден пакет за достигане на неговото местоназначение.

Такива команди за следните

Tracert – за Windows

Traceroute - за Linux

Други TCP/IP помощни програми

Тестването на възможността за връзка и проверката на информацията за конфигурацията са най-честите приложения на TCP/IP помощните програми при отстраняване на проблеми в мрежата. Но наред с тях могат да бъдат използвани и няколко допълнителни инструмента за събиране на конкретна информация:

- **Netstat** и **Nbtstat** - Те визуализират TCP/IP и NetBIOS статистическа информация.

- **ARP** (или „arp“ в Linux и UNIX) - Използва се за визуализация и управление на кеша на протокола Address Resolution Protocol (ARP).

- **ROUTE** (или „route“ в Linux и UNIX) - Използва се за преглеждане и промяна на елементите, записани в маршрутните таблици.

Литература

1. Боянов, К. Локални компютърни мрежи, С., Техника, 1989.
2. Комър Брайън. TCP/IP мрежи и администриране. С., ИнфоДар, 1999.
3. Кочан, С., Ууд П.. Запознаване с операционната система UNIX, С., Paraflow, 1993.
4. Мерике Каео. Проектиране на мрежова сигурност. С., СофтПрес, 2006.
5. Николов, Л. Операционни системи, С., Сиела, 1998.
6. Нортън, П. Пълно ръководство за Linux, С., ИнфоДар, 2000.
7. Нортън, П., Д.Кърнс. Пълно ръководство за работа с мрежи, С., ИнфоДар, 2000.
8. Остерло Хедър. TCP/IP пълно ръководство. С., СофтПрес, 2002.
9. Шиндър Д. Компютърни мрежи. Пълно ръководство по теория, изграждане и съвместна работа между мрежите. С., СофтПрес, 2003.
10. Швета Базин. Основи на мрежовата сигурност. С., ДуоДизайн, 2004.
11. Пашова, М. Съвременни системи за достъп, София, БТК, 1999.
12. Симеонов, Ст., Катъров П., Съвременни компютърни комуникации. Принципи и реализация, Бургас, АПН, 2001
13. Comer, D.E. Internetworking with TCP/IP, Vol.1. Principles, Protocols and Architecture, Englewood, Prentice Hall, 1995.
14. Comer, D.E. Internetworking with TCP/IP, Vol.11. Design, Implementation and Internals, Englewood, Prentice Hall, 1996.
15. International Organization for Standardization, Information Processing Systems - Open Systems Interconnection - Connection Oriented Transport Protocol Specification, ISO 8073, 8824,8825, 9595,9596, ISO Publishing House, Switzerland.
16. CISCO Network Module Hardware Installation Guide, CISCO Systems, 2000.
17. Software Configuration Guide, CISCO Systems, 1999.
18. David, S.A. Inside Windows NT - 2nd edition, Microsoft Press, 1998.
19. Hunt, C. TCP/IP Network Administration, Second Edition, December 1997
20. Garfinkel, S., G. Spafford. Practical UNIX and Internet Security, Second Edition, O'Reilly, April 1996
21. Peek, J., T.O'Reilly, M. Loukides. UNIX Power Tools, August 1997
22. Liu, C, P.Albitz. DNS and BIND, O'Reilly, September 1998
23. International Technical Support Organization of IBM, TCP/IP Tutorial and Technical Overview Rep.GG24-3376-05, October 1998
24. CISCO Systems, Internetwork Design Guide, 2000.
25. CISCO Systems, Internetworking Technology Overview, 2000.
26. Troubleshooting and Configuring the Windows NT/95 Registry, Macmillan Computer Publishing,
27. Deering, S.E., D.R.Cheriton. Multicast Routing in Datagram Internetworks and Extenden LANs, ACM Transactions on Computer Systems, 8(2), 1990, pp.85-110.
28. Falk, G. The Structure and Function of network Protocols, in Computer

- Communications, vol.1, Cheu, W(ed.), Englewood, Prentice Hall, 1983.
29. Rose, M. (ed.) Management Information Base for Network Management of TCP/IP Based Internets, DDN network Information Center, SRI International, Ravenswood (USA).
 30. Karn, P., C.Partridge. Improving Round-Trip Estimates in Reliable Transport Protocols, Proc. ACM SIGCOMM'87.
 31. Martin, J. Computer Networks and Distributed Processing, Englewood, Prentice Hall, 1991.
 32. Comer, D.E. Internetworking with TCP/IP, Vol.11. Design, Implementation and Internals, Englewood, Prentice Hall, 1996.
 33. Comer, D.E., D.Stevens. Internetworking with TCP/IP, Vol.III. Client-Server Programming and Applications, Englewood, Prentice Hall, 1996
 34. Comer, D.E. The InternetBook: Everything you need to know about computer networking and how the Internet works, Englewood, Prentice Hall, 1995
 35. Comer, D.E., D.L.Stevens. Vol.III, Windows Sockets Version, Englewood, Prentice Hall, 1997.
 36. Denning, Dorothy E. *Information Welfare and Security*. Reading, MA: Addison-Wesley, 1999 r.
 37. Kaufman, C, R. Perlman , M. Speciner. Network Security: Private Communication in a Public World, Второ издание. Upper Saddle River, NJ: Prentice Hall PTR, Ё 2002 r.
 38. McCarthy, Linda. *Intranet Security: Stories from the Trenches*. Palo Alto, CA: Sun Microsystems Press, 1998 r.
 39. Pfleeger, Charles, et al. Security in Computing, Трето издание. Upper Saddle River, NJ: Prentice Hall PTR, 2002 r.
 40. Rescola, Eric. SSL and TLS: Designing and Building Secure Systems. Reading, MA: Addison-Wesley Professional, 2000 r.
 41. Schneier, Bruce. Applied Cryptography, Второ издание. New York, NY: John Wiley and Sons, 1996 r.
 42. Stallings, William. *Cryptography and Network Security*, Трето издание. Upper Saddle River, NJ: Prentice Hall, 2002 r.
 43. Chapman, D. Brent ,Elizabeth D. Zwicky. Building Internet Firewalls, Второ издание. Cambridge, MA: O'Reilly and Associates, 2000 r.
 44. Chapman Jr., David W , Andy Fox. Cisco Secure PIXFirewalls. Indianapolis, IN: Cisco Press, 2001 r.
 45. Cheswick, William , Steven Bellovin. Firewalls and Internet Security, Второ издание. Reading, MA: Addison-Wesley, 2002 r.
 46. Carter, Earl. Cisco Secure Intrusion Detection System. Indianapolis, IN: Cisco Press, 2001 r.
 47. Northcutt, Steven , Judy Novak. Network Intrusion Detection: An Analyst's Handbook, Трето издание. Indianapolis, IN: New Riders, 2002 r.