ПЛОВДИВСКИ УНИВЕРСИТЕТ „ПАИСИЙ ХИЛЕНДАРСКИ“
ФАКУЛТЕТ ПО МАТЕМАТИКА, ИНФОРМАТИКА И ИТ
КАТЕДРА „КОМПЮТЪРНИ СИСТЕМИ“

Информационна и комуникационна сигурност

Изготвил: доц. д-р Генчо Стоицов

Пловдив, 2020 г.

Лекция 1

Мрежовата сигурност

Има няколко ключови моменти, свързани с мрежовата сигурност:

- Колко сигурни са системите, контролиращи обмена на информация по мрежата?
- Колко сигурна е информацията, съхранявана на многобройните компютри в мрежата?

Известно е, че с всичко, което може да се използва, може и да се злоупотребява. Сигурността на системно ниво означава системата да не може да бъде пробита до такова ниво, че да се разпадне. Сигурността на ниво данни означава данните да не могат да бъдат подменяни.

Провал в мрежовата сигурност може да струва много от гледна точка на репутацията. Никоя организация няма да се интересува от работа с компания, която не може да запази информацията и сигурността на системата си.

Следователно:

- С разрастването на използването на интернет вероятността за злоупотреба с него нараства. Шансът за пробив в сигурността е много висок, ако мрежата на организация не е добре защитена.
- Важно е да се планира сигурността на организационно ниво.
- Една дупка в сигурността струва на компания не само пари, но и добра репутация.
- Базите данни съхраняват поверителна информация. Затова е важно критичната информация да се предпази от злоупотреби.
- Колкото по-сложна е една система, толкова по-предразположена към пробиви и грешки е тя. Отношението към сигурността поради незнание трябва да бъде избегнато.

Какво е пробив в сигурността на мрежата?

Пробивът в сигурността може да се дефинира като нелегален достъп до информация, който може да причини разкриване, заличаване или смяна на информацията. С други думи, пробив в сигурността означава използване

или достъп до информация или системи за незаконни цели или за цели, за които информацията или системите не трябва да се използват.

Пробивите в сигурността могат да се проявят в няколко различни форми - посегателство над мрежата на организация, пробиване на системи или мрежи, смяна на организационна или индивидуална информация, атаки с вируси, спиране или отхвърляне на услуги, счупване и кражба.

Някои често срещани типове пробиви:

Достъп до информация на абонати за изпращане на спам съобщения

Неоторизиран достъп до база данни на доставчик на услуги и използването им за промоция на нови оферти от трета страна.

Неупълномощен достъп до поверителна информация с цел създаване на подправена самоличност

Човек получава достъп до информация като местожителство, телефони за контакти, осигурителен номер и номер на сметка от база данни на банка и с тази информация може да си създаде фалшива самоличност.

Подслушване

Разузнаването на някоя страна се свързва с мрежа на друга страна и може да придобие поверителна информация за отбраната ѝ.

Използване на автоматизиран скрипт с цел влизане в компютърна система

Хакерите използват автоматизирани скриптове, за да направят редица опити да влезнат в компютърна система. В резултат компютърът отказва достъп на упълномощените потребители, защото е зает да отказва заявките на хакера.

Придобиване на неупълномощен достъп до мрежа с цел достъп до фирмена информация

Неправомерен достъп до мрежа на организация може да е предпоставка за кражба на поверителна фирмена информация.

Вирусни атаки

Вирусите обикновено се разпространяват чрез е-поща. Вирусна атака може също да започне от мрежата на организация и през нея да се разпространи в Интернет.

DNS превземане

Domain Name System е база данни, която задава връзката между имената на доменните и IP адресите. Компютри, които са свързани с Интернет, използват DNS, за да превърнат URL адрес (имената на сайтовете) в IP адреси на сайтовете, които искат да отворят. При DNS превземане хакерът има достъп до DNS услугите и променя информацията, която задава връзката между името на домейн и IP адрес. Заради това потребителите биват пренасочвани към различен сайт от този, който са искали.

DoS (denial of service) атаки

DoS атаките (атаките от тип “отказ на услуга”) са мрежово базирани атаки, при които на упълномощени потребители се отказва достъп до мрежови услуги. Причините за DoS атаките са различни, например неупълномощен достъп до ресурси.

DDoS атаки

Разпределените DoS атаки са усъвършенствани форми на DoS атаките. В DDoS атаките системата, която е мишена, е атакувана с няколко различни компютъра през Интернет. Без знанието на собственика, хакерът създава приложение и го поставя на многобройни места в Интернет. Такива приложения не се разпознават, тъй като не вредят на системата, в която се намират. Когато атаката започне, мишената е атакувана от всички компютри, на които е инсталирано приложението.

DDoS атаките са трудни за разкриване, тъй като атаката не идва от един източник. Ако без знанието на собственика някой компютър бъде използван за атаки над други мрежи, собственикът става компрометиран.

Собствениците на системата-мишена могат да предприемат законни действия срещу компрометираната система. Причината за това е, че се очаква от всяка система в мрежата да пази ресурсите си от злоупотреби. Ако

някоя система се провали в собствената си защитата, то тя е отговорна за атаките, без значение, че не е участвала съзнателно в тях.

Пробив във физическите граници на организация

Разбит е офис на организация и са откраднати твърдите дискове на сървъра с базата данни.

Кое в глобалната мрежа има нужда от защита?

За осигуряване на сигурността на глобалната мрежа, е необходимо да се разбере точно кое в нея има нужда от защита. Независимо дали става дума за един човек или за цяла организация, сигурността по мрежата най-вече означава сигурност на информацията. Три компонента в глобалната мрежа е необходимо да бъдат защитени:

- клиентът;
- сървърът;
- мрежата.

Тези компоненти си взаимодействат по следния начин в Web архитектурата:

1. Клиентът изпраща заявка до сървъра чрез въвеждане на URL адрес в Web браузъра.
2. По подразбиране заявката първо се насочва към DNS сървъра. DNS сървърът преобразува името на домейн в URL адреса в IP адреса на Web сайта, който клиентът е поискал.
3. Чрез използване на IP адреса се създава връзка със сървъра, на който се намира Web сайтът и се изпраща заявка до Web сървъра за Web страницата.
4. Web сървърът изпраща исканата Web страница до клиента.
5. Браузърът показва Web страницата на екрана. Това взаимодействие между клиента и сървъра е улеснено от няколко мрежи, които свързват всички компютри в глобалната мрежа.

Заплахите в сигурността, които всеки от изброените компоненти среща в средата на глобалната мрежа са:

Заплахи за клиента

Компютърът от страната на клиента е уязвим към вирусни атаки от хакери, кракери и злонамерен код. Освен това клиентът в глобалната мрежа е предразположен към посегателство над личната си информация и самоличност.

Заплахи за сървъра

Данните на Web сървърите са уязвими за неупълномощен достъп. Възможни са нарушения в сървъра, които да доведат до намаление на скоростта му на работа и в най-лошите случаи — до спирането му. Освен това ресурсите на сървъра могат да се използват с различна цел от тази, за която са предназначени.

Заплахи за мрежите

Една мрежа в Web, ако не е защитена правилно, би могла да се превърне в същинската причина за посегателство над информация. Причината затова е, че мрежата е входна точка за компютърните системи. Слабата мрежа позволява при пренос на данни от компютър-източник до компютър-получател информацията да бъде променена и фалшифицирана. Хакерите могат още да използват ресурсите на компютъра чрез пробив в мрежата. Повечето случаи на фалшива самоличност и подслушване са възможни поради дупки в сигурността на мрежата.

Аспекти на мрежовата сигурност

При осъществяването на мрежовата сигурност трябва да се обмислят два важни аспекта. Първият включва дефиниране на целта на сигурността. Вторият включва създаване на уравнение на сигурността. Уравнението на сигурността е свързано с установяване на критичната информация, която трябва да бъде защитена, и на какво ниво на защита.

Цел на мрежовата сигурност

Мрежовата сигурност е необходима, за да осъществи:

- поверителност
- цялостност
- достъпност

- проверка за автентичност

Поверителност

Терминът поверителност (конфиденциалност) се отнася до защита на важна информация от разкриване за неупълномощени потребители. Степента, до която трябва да бъде поддържана поверителността, зависи от типа информация, която трябва да се защити.

Например изисква се високо ниво на поверителност, когато годишният доклад на организация (който още не е публично огласен) се препраща по мрежата. Обратно, нивото на поверителност може да е сравнително ниско, когато се препраща докладът за печалбата от предното тримесечие (който вече е бил публично огласен).

Друг пример, който илюстрира разликата в нивата на поверителност, които се поддържат, е типът информация, която организацията споделя със служителите си, и обратно - с хора извън организацията. Една организация би споделила всички свои политики и процедури със служителите си и същевременно тя не би желала да споделя такава информация с външни хора. По същия начин информацията за оценките на служителите ще бъде достъпна само за супервайзорите и за отдела "Човешки ресурси", а до такава информация няма да имат достъп всички служители на компанията.

Посегателството над поверителността може да има отрицателно влияние върху организацията. Все пак ефектът от една такава вреда зависи от важността на разкритата информация. Затова трябва да се поддържат различни нива на поверителност за различните типове информация. Например, ако конкуренцията разкрие подробности за бъдещи клиенти и оферти, направени от организацията към тях, това може да причини сериозни загуби. Също така, ако информация за оценките на всички служители бъде разпространена по някакъв начин помежду им, това може да причини недоверие и смут между тях.

Цялостност

Терминът цялостност на данните означава гарантираност, че данните не са променяни от неупълномощени потребители. Важно е да се поддържа целостта на информацията в мрежата, ако не искате тя да бъде изтълкувана погрешно от желаните потребители. Следват няколко често срещани начина, по които се нарушава целостта на данните:

- Промяна на доклад от финансова ревизия. Докладите са важни за акционерите, клиентите, ръководството и служителите. Ако някой неупълномощен потребител промени доклада, той не би създал верен образ на организацията.
- Промяна на банкови сметки. Ако бъде променен текущият баланс в спестовна сметка с по-голяма сума, банката може да понесе тежки загуби.
- Промяна на съдържанието на новинарски сайт. Ако бъде променено съдържанието на новинарски сайт и бъдат публикувани измислени новини, това може да причини безредие и безпокойство у читателите.

Нарушаването на целостта на информацията не винаги е умишлено. По време на въвеждането и съхраняването на данните могат да се допуснат грешки. Например операторът, въвеждащ данни, може да напечата грешни числа, които да превърнат загубите на компанията в ползи. Целостта на данните може да се загуби, ако файловете или системите са развалени или напълно унищожени.

Достъпност

Осигуряването на достъпност означава данните или информацията да бъдат налични за използване при необходимост. В контекста на мрежовата сигурност това по същество означава поставяне на системите за сигурност на такива места, които да предотвратят неупълномощени действия в резултат на недостъпността на информацията. Терминът достъпност не е ограничен до достъпност на информация, той включва достъпност на системите и останалите ресурси, необходими за достъп до информацията.

Освен това осигуряването на достъпност на информацията включва подсигуряване, че потребителите не са лишени от услуги, когато те са необходими. Отказът на услуги (DoS) се случва, когато системата е затрупана от заявки от неупълномощени потребители и като резултат упълномощените потребители не получават достъп.

Проверка за автентичност

За избягване на нахлуване на хакери в система или мрежа, е важно да се проверява автентичността на потребителите. Проверката за автентичност (автентикацията) е механизъм за потвърждаване, че опитващият се да

достигне до ресурс в глобалната мрежа е този, за който се представя. Проверката също така включва гарантиране, че само упълномощени потребители имат достъп до информацията.

Уравнение на сигурността

Създаването на уравнение на сигурността е важен аспект при планирането и изпълняването на критериите на сигурността. В основни линии това уравнение е част от процеса на управление на риска. Управлението на риска включва главно определянето на необходимата защита, както и на съответните мерки за сигурност, които трябва да бъдат изпълнени по ефективен начин спрямо цената. Определянето на уравнението на сигурността налага строг анализ в контекста на цената, включена в изпълняването на мерките за сигурност, и очакваната печалба от тези мерки.

Целта на създаването на уравнение на сигурността трябва да е осъществяване на баланс между разходите и печалбата. Това означава, че трябва да се отдели допълнително внимание при определянето на цената за осъществяването на мерки за сигурност. Организациите трябва да постигнат същата или дори по-голяма печалба от тази цена.

Трябва да се имат предвид следните фактори при създаване на уравнението на сигурността:

- реалната стойност на информацията;
- възприетата стойност на информацията;
- цената на защитата на информацията;
- цената на пробив в защитата.

Реалната стойност на информацията

Реалната стойност на информацията означава действителната стойност на информацията, която организацията иска да защити. Определянето на действителната стойност не е лесна задача, тъй като за разлика от материалните вещи, не винаги е възможно да се определи фиксирана финансова стойност на информацията. Цената на информацията за една организация е винаги относителна и зависи от различни фактори. Следват няколко фактора, които определят стойността на информацията:

- Пазарната цена на информацията или цената, на която информацията може да бъде продадена на пазара.
- Цената за достигането до информацията.
- Цената за възстановяване на информацията (в случай че е загубена).
- Цената на операциите, включително цената на всички операции, засегнати от загубата на информацията.

Възприетата стойност на информацията

Възприетата стойност на информацията се отнася до печалбата, която потребителят ще извлече от достигане до информацията по нечестен начин. Например, ако служител продава важна информация за организацията на конкуренцията. В този случай възприетата стойност на информацията ще бъде сумата пари, която служителят получи за информацията.

Също както действителната стойност, възприетата стойност зависи от някои фактори. Един от тях е начинът, по който се използва продадената информация. С други думи, възприетата цена е различна за различните потребители.

Друг пример: някой краде софтуер за системи за обработване на транзакции, направен по поръчка на организация. Ако откраднатият софтуер бъде продаден на подобна организация, възприетата му цена ще е висока, тъй като конкурентната организация може да спести парите, които в противен случай ще използва за създаване на софтуера. От друга страна, възприетата цена на информацията няма да е много висока, ако софтуерът бъде продаден на организация, която няма да го използва.

Цената на защитата на информацията

Цената на защитата на информацията включва всички финансови и нефинансови ресурси, инвестирани в осъществяване на мерки за защита на информацията. Нефинансовите ресурси включват времето и усилието, прекарани в реализиране, поддръжка и контрол на системите за сигурност. Цената на разработването и поддържането на контролите за сигурност трябва да е по-малка или равна на цената на информацията, която защитават.

Цената на пробив в защитата

Цената на пробив в защитата включва цената, която нарушителят ще трябва да плати, за да влезе в системата и да си осигури достъп до информацията. Например, за да достигне до някоя важна информация една организация може да наеме хора да пробият системата за сигурност на друга организация. Така цената на пробива ще включва заплатата, която организацията плаща на тези хора, инструментите, необходими за пробиване на системата, и цената на времето, прекарано в пробиване на системата.

Дефиниране на уравнението на сигурността

При дефиниране на уравнението на сигурността една организация трябва да има предвид следните фактори:

- Организацията трябва да оцени стойността на активите (тук става дума за информацията - действителна и възприета). След това трябва да оцени цената на съществуващите контроли за сигурност в сравнение с тази на нови контроли (в случай, че планира да реализира нови такива). Тази задача може да бъде изпълнена чрез назначаване на екип за управление на риска.
- Организацията трябва да е сигурна, че цената на защитата на информацията е по-малка или равна на действителната цена на информацията. Това означава, че стойността на ресурсите, използвани за защита на информацията, няма да надвишават загубата, която организацията може да понесе, ако защитата на информацията бъде пробита.
- Цената на пробиване на информацията трябва да е по-голяма от възприетата цена на информацията. Това означава, че цената, която нарушител ще плати, за да пробие системата за сигурност, трябва да е по-голяма от печалбата, която ще получи при продажбата ѝ.
- Подсигуряването да бъде такова, че цената за пробива на приложените защитни контроли да е по-висока от цената на информацията, придобита при пробива на тези контроли. Този принцип трябва да поддържа сигурността. Например, дали един хакер би изхарчил 100 долара, за да получи информация, която струва само 10 долара?

Лекция 2

Заплахи и уязвими страни

В контекста на сигурността на информационните технологии заплахите могат да се дефинират като вътрешни или външни действия или събития, които могат да причинят вреди на системите, приложенията или информацията на организация. Вътрешни или външни означава, че опасността може да идва отвътре или извън организацията. Това може да причини неупълномощено разкриване, преместване, разрушаване или унищожаване на информация и системи. Обикновено дефиницията за заплахи е “всякакви обстоятелства или събития, които могат да навредят на система или мрежа”.

Заплахите могат да се класифицират като физически или електронни. Физическите заплахи увреждат машините и връзките (унищожаване на съоръженията за комуникации). Електронните заплахи идват от страна на хакерите чрез злонамерени приложения под формата на вируси, червеи, троянски коне или атакуващи скриптове. Електронните заплахи могат да причинят по-големи вреди от физическите.

Заплахите съществуват, когато има уязвими страни (уязвимости) в система или мрежа. Понякога няма уязвимости, но системата е винаги открита за заплахи.

Уязвимостта е пробойна или вратичка в система или програма, която улеснява хакера при опита му да влезе в система за сигурност. Дефинира се като “аспект от система или мрежа, който я оставя открита за атака”.

Уязвимостите биват известни и неизвестни. Известните уязвимости са тези, за които се знае по времето на производството. Допускаме например, че има пробойни в операционната система, използвана от някоя организация. Въпреки това организацията не знае за тях и продължава да използва тази операционна система без да вземе предпазни мерки. Заради това организацията е изправена пред различни заплахи. Нарушители, които знаят за дупките в операционната система, се опитват да извлекат максимална полза от това и причиняват увреждане на информацията и системите. Това е типичен пример за известна уязвимост. Повечето известни уязвимости водят до заплахи за сигурността, защото организациите не се придържат към приетите практики за защита.

Неизвестните уязвимости са тези, които съществуват в системите, но не са засечени от производителите. Например организация използва операционна система, в която има дупки, които не са известни нито на организацията, нито на производителите. Нарушител, който се опитва да пробие мрежата на организацията, попада на бъговете и през тях намира начин да достигне до информацията на организацията. В този случай, ако нарушителят е незлонамерен хакер, той може да информира организацията за дупките в системата, но ако е с лоши намерения, може да извлече максимална полза от пробойната и да изложи на опасност системата на организацията.

Типове заплахи

Глобалната мрежа е предразположена към много заплахи, включително компютърна измама, шпионаж, вандализъм, заплаха за репутацията, компютърни вируси и други хакерски опити. С нарастването на доверието на света към глобалната мрежа заплахите стават все по-разпространени и сложни. Базирайки се на същността и източника на атаките, те се класифицират в следните основни категории:

- случайни заплахи;
- злонамерени заплахи;
- заплахи за пълномощията;
- заплахи за приложенията;
- заплахи за поверителността;
- заплахи за контрола на достъпа.

Всяка от горните категории обхваща няколко подгрупи, които са разгледани по- долу.

Случайни заплахи

Както предполага и името, случайните заплахи обхващат непланираните и неумишлени заплахи. Те покриват проблеми, които предимно са резултат от човешка грешка като: лош избор на пароли, случайни или погрешни бизнес транзакции, случайно разкриване на информация и използване на неподходящ или остаря софтуер. Случайните заплахи основно са резултат от недостатъчни познания в системите за

сигурност, неправилната конфигурация на устройствата за сигурност и изтичане на информация поради незащитен трансфер на данните.

Злонамерени заплахи

Злонамерените заплахи целят специално причиняването на вреди на хора, системи и мрежи на организацията. Можем да ги разделим на следните подгрупи:

- злонамерен софтуер;
- неправомерно събиране на информация (social engineering - социално инженерство).

Злонамерен софтуер

Злонамереният софтуер е програмен код, който е целенасочено написан, за да причини вреда на система или мрежа. Вирусите, червеите и троянските коне са типични примери за злонамерени програми.

Вируси

Компютърният вирус е злонамерена програма, която е устроена да се прикрепя към друга програма, файл или дори сектора за зареждане на твърдия диск. Подобно на вирусите в биологията, компютърните вируси се нуждаят от приемник (хост). Обикновено такъв приемник се явява някой изпълним файл или офис документ, който бива заразен. Когато този файл бъде отворен, вирусът се изпълнява и може да порази много други файлове в системата.

Действието на вирусите може да бъде различно. Някои от вирусите са по-безобидни – тяхното действие се свежда до извеждане на съобщения на екрана на компютъра. Съществуват и вируси, които са доста по-злонамерени – те могат да повреждат, променят и изтриват файлове от компютърната система, така че компютърът да не може да бъде стартиран. Например известният вирус Чернобил изтрива съдържанието на EEPROM FLASH BIOS паметта, с което го увреждa.

Обикновено вирусите се класифицират по видове. Най-често срещаните от тях са:

- BOOT-секторни – вируси, заразяващи сектора за първоначално зареждане (boot record);
- BIOS вируси - заразяват входно-изходната система на компютъра;
- Файлови вируси - заразяват всички активни програмни файлове (*.COM, *.EXE, *.OVR, *.BIN, *.SYS);
- Стелт – те не променят размера на заразения файл. Секторите, които заразяват се маркират като лоши, въпреки че не са повредени;
- E-Mail вируси – особено актуална категория вируси. Разпространяват се чрез електронна поща и използват адресната книга, за да нападат нови компютри.

Троянски коне

Троянските коне са програми, отварящи вратичка в сигурността на системата. Те дават неоторизиран достъп на атакуващия компютър, намиращ се в Интернет. Атакуващият може да вижда съдържанието на екрана, да стартира и спира приложения, да изтегля и изпраща файлове, да изтрива файлове, да форматира дискове, да спира или рестартира компютъра и т.н.

Троянският кон е изпълним файл и заразяването става винаги чрез стартирането му от потребителя. Троянският кон обикновено е програма от две части – клиент и сървър. При стартиране на програмата сървър се отварят един или повече порта на заразения компютър. Чрез клиентската програма злонамерен потребител може да получи отдалечен достъп до този компютър.

Съществуват програми – троянски коне, които се представят като други програми. Това може да бъде например прозорец за логване в системата. При въвеждането от потребителя на името и паролата, данните се предават по мрежата и съответно този, който е стартирал програмата-клиент от троянския кон може да получи достъп до системата.

За разлика от вирусите, троянските коне не се самокопират.

Червеи

Червеите са програми, които се самокопират от компютър на компютър, но не заразяват други програми. Основната вреда от тях е от гледна точка на процесорното време и часовете, загубени в опити да ги отстраните от системата си. При разпространяването си се изпращат на части и след това се “сглобяват”- това се прави с цел да бъдат по-трудно откриваеми. Първо пристига стартиращ модул (Starter) – малка програма изтегляща и реконструираща червея. Преминаването само на малки фрагменти от информация през входно-изходната система не позволяват на антивирусните програми да разпознаят червея.

Понятието „червей” много често се използва като синоним на „вирус”, но между двете има известна разлика. По принцип, червеите се смятат за подклас на вирусите, но за разлика от тях те могат да се пренасят от един на друг компютър без помощ от страна на човека. Това не асистирано пътуване е възможно поради факта, че те се възползват от транспортните функции на самите файлове или самата информация.

Опасното при червеите е, че те могат да се възпроизвеждат, което означава, че компютърът ни може да изпрати в пространството стотици или хиляди копия на първоначално влезлия в системата му червей. Много често компютър, заразен с червей, изпраща копия на този червей до всички потребители вписани, например, в адресната книга в имейл акаунта на собственика му. Ако вирусите модифицират различни файлове на заразения компютър, то компютърният червей може да не засегне нито един от тези файлове. За сметка на това, червеят може да стои зареден в системната памет на заразения компютър и там се репликира, търсейки пътища към нови компютри.

Поради това, че може да се възпроизвежда и да пътува безпрепятствено, червеят започва да консумира голямо количество системна памет или мрежова лента и може да спре работата на уеб сървъри, мрежови сървъри или индивидуални компютри.

Примери:

- Преди време широкоразпространен червей на име Blaster Worm успява да проникне в системата на многобройни потребители,

правейки я уязвима за хакери. С достъп до нея, тези хакери можеха да управляват засегнатите компютри от разстояние.

- През 2017 година червеят WannaCry (WanaCrypt0r 2.0) причинява невиждани щети за милиарди по цял свят. WannaCry е познат повече като рансъмуер атака, но той всъщност е хибрид между червей и рансъмуер. Рансъмуерът (ransomware) е вид злонамерен софтуер, който прониква в системата и ограничава достъпа до потребителските файлове, като ги криптира или заключва системата. Това обикновено е придружено от съобщение с искане на откуп, често пъти в криптовалута, след заплащането на който потребителя би трябвало да получи скрипт за „отключване“ или декриптиране на своите данни. WannaCry се възползва от уязвимостта на протокола Microsoft Server Message Block ver.1 (SMBv1), който се използва за споделяне/достъп до файлове в някои мрежови устройства. Червеят използва EternalBlue експлойт, който използва въпросната уязвимост в SMB протокола. Така той инсталира „задна вратичка“ в системата, която позволява изтеглянето на софтуера. Червеят поразява множество Windows-базирани системи, които използват стара версия на операционни системи Windows, както и системи, на които не са правени ъпдейти за сигурност. WannaCry може да се самовъзпроизвежда и веднъж попаднал в даден компютър може да зарази всички останали, свързани в мрежата.
- През 2010 година първият компютърен червей, използван като кибер-оръжие беше открит след поредица инциденти в Иран. От Symantec съобщават, че червеят е създаден с цел атака на иранска електроцентрала, а крайната цел е да се саботира производството на ядрено оръжие. Този червей, наречен Stuxnet има изключително сложен механизъм, невиждан до този момент. Stuxnet атакува определени контролери на Siemens, които се използват широко в индустрията, включително и в иранската ядрена програма.
- Sasser & Netsky са два отделни червея, често групирани заедно поради сходствата си в кода. Именно заради тези прилики се смята, че червеите Sasser и Netsky са създадени от един и същи човек – 17-годишния германски студент Sven Jaschan. Sasser се разпространява чрез заразени компютри, като сканира случайни

IP адреси и ги инструктира да изтеглят зловредния код. Netsky от своя страна използва по-познатия подход за разпространение чрез използването на електронна поща. Потребителите биват приканени да отворят прикачен към писмото файл, съдържащ зловреден код. След като този файл бъде стартиран програмата започва да сканира компютъра за e-mail адреси и изпраща свое копие до всеки открит електронен адрес.

- ILOVEYOU е друга голяма атака от червей, нанесла огромни щети на десетки милиони компютри. Атаката започва през май 2000 година. Червеят се разпространява чрез електронната поща, под формата на съобщения, носещи заглавието „ILOVEYOU“. Към въпросните съобщения е прикачен файл с „двойно разширение“, носещ името „LOVE-LETTER-FOR-YOU.txt.vbs“. Повечето Windows-базирани мейл клиенти скриват истинското разширение (.vbs) на файла и потребителите биват заблудени, че това е текстов (.txt) файл. Отваряйки въпросния файл се изпълнява Visual Basic скрипт, който инфектира системата. Веднъж активирал се, ILOVEYOU презаписва както системни файлове, така и лични файлове – офис документи, музика, изображения и т.н. След това червеят започва да се самовъзпроизвежда отново и отново.

Най-често използваните начини за заразяване на компютърните системи с червеи включват:

Чрез електронната поща

Най-често това става като червеят пристигне като прикачен към съобщението файл или интегриран в самия текст на писмото. Когато потребителят отвори писмото или файла, прикачен към него, червеят прониква в системата. След това той може да започне да създава свои копия, които да изпрати до наличните контакти в адресната ви книга.

Чрез уязвимости в операционната система, инсталирания софтуер и дори в мрежовите протоколи

Ето защо е много важно да обновяваме операционната система с всички налични ъпдейти за сигурност, както и да използваме защитна стена.

Чрез съобщения

В миналото софтуерът за мигновени съобщения като mIRC, MSN Messenger, Yahoo IM и ICQ се оказа изключително плодотворна среда за разпространение на компютърни червеи. Днес използването на FB Messenger, Skype и други подобни платформи също крие известни рискове, особено ако се доверяваме на всеки получен като съобщение файл.

Заплахи тип социално инженерство

Защитата на компютрите в глобалната мрежа наподобява създаването на щит около мрежите, който да ги пази от заобикалящия ги свят. За съжаление, независимо от количеството пари и време, които загубим в защита на мрежите, винаги забравяме един фактор - човешката психология. Служейки си с техника, наречена социално инженерство, много атакуващи и хакери заобикалят и най-строгите системи за защита.

Социалното инженерство може да се дефинира като изкуството да използвате външни умения, за да извлечете поверителна информация. То по същество включва външно лице, което лъже персонала на организация да му осигури частна информация или нелегален достъп до ресурси.

Социалното инженерство е дефинирано от много автори. Една от тези дефиниции е следната:

“Социалното инженерство може да бъде разгледано като ‘хакване на хора’, по същество хакерски жаргон за настоятелно търсене на несъзнателно участие от човек в компанията, вместо самостоятелно пробиване на системата.”

Основни техники, използвани в социалното инженерство

Социалният инженер може да приложи множество техники, а тяхното влияние зависи от способностите на инженера и от умението му да убеждава хората. Добрият социален инженер може да започне с проучване на организацията- мишена (или просто мишената), за да добие идея за нейната основна структура и имена- на служителите. Придобитата от това проучване информация може да не помогне веднага на нарушителя, но може и да му бъде от полза по-нататък, за да получи друга информация. Тогава нарушителят може да използва различна техника, да постигне целта си.

Всяка техника, използвана в социалното инженерство може да се класифицира в две категории: *атаки, базирани на хора*, и *атаки, базирани на компютри*.

Атаките базирани на хора използват лични способности, връзки и измами за получаване на информация. Този вид атаки е тест за инженера от гледна точка на това колко добре може да ласкае, изнудва или лъже.

Например нарушител, представящ се за клиент на банка, се обажда на банков служител. Нарушителят съчинява история и рано или късно пита служителят за номера на сметката и паролата. Служителя невинно дава информацията, мислейки нарушителят за истински клиент. Той осъзнава грешката си само тогава, когато истинският клиент докладва за злоупотреба с парите в сметката.

Атаките, базирани на компютри използват в основата си технологии и мамят хората да предоставят информация.

Например в глобалната мрежа потребител може да срещне прозорец със съобщение "Връзката с мрежата е загубена. Моля въведете вашето потребителско име и парола, за да се свържете отново". Веднъж, след като паролата е попълнена, информацията се изпраща на нарушителя.

Често използваните техники на социалното инженерство:

- *Директен подход* - социалният инженер може директно да помоли мишената за информация. Пример за това е обаждане до служителя на рецепцията (рецепциониста) на организация с въпрос за имената и паролите на потребителите. В повечето случаи този подход не е успешен, тъй като днес хората мислят за сигурността и внимават при предоставянето на такава информация.
- *Важен служител* - друг подход на социалното инженерство е нарушителят да се представи за старши служител на организация. Обикновено подчинените или служителите с по-ниски чинове се подчиняват на старши служителите в организацията. Използвайки това, нарушителят притиска подчинените с налагане на важни крайни срокове, като по този начин успява да получи важна информация относно софтуера за отдалечен достъп в компанията, начините да го конфигурира, телефонните номера

на сървъра за отдалечен достъп (Remote Access Server, RAS), както и името и паролата, използвани за достъп до него. След получаването на тази информация нарушителят може да установи отдалечен достъп до мрежата на организацията.

- *Нов и безпомощен служител* - при тази техника нарушителят се представя за служител, който има нужда от помощ за достъп до ресурсите на организацията. Този тип атака е прост. Атакуващият се обажда на секретарката на организацията, представяйки се за нов служител, който има проблеми с достъпа до мрежата. Секретарката в нежеланието си да го обиди или да изглежда некомпетентна може да е склонна да му помогне, като му даде потребителско име и парола на съществуващ акаунт или дори на собствения си.
- *Техническа поддръжка* - нарушителят се представя за член на екипа за техническа поддръжка на организацията и извлича информация от невинни и невежи служители, като се преструва на системен администратор, който се опитва да реши мрежови проблем. За да успее, той пита за потребителско име и парола.
- *Обърнато социално инженерство (Reverse Social Engineering, RSE)* - типична RSE атака се състои от 3 части - саботаж, рекламиране, асистиране. Нарушителят саботира работното място на потребител или му придава вид на развалено. Виждайки това, потребителят търси помощ. С цел да се подсигури, че потребителят ще се обади точно на него (а не упълномощения технически персонал), нарушителят рекламира присъствието си или с оставяне на визитната си картичка на работното място на служителя, или с показване на номера си на самото съобщение за грешка. Накрая нарушителят асистира в решаването на проблема и по този начин получава исканата информация.
- *Измами по e-mail* - измамата по e-mail включва използването на актуална тема, за да предизвика емоции, които могат да доведат до несъзнателно участие от страна на потребителя. Например потребителят може да получи e-mail съобщение от почтен Web сайт, който разисква чувствителни въпроси за детския труд. Сайтът обещава да плати парична награда, ако потребителят отговори на няколко въпроса. Освен това го моли да попълни формуляра, прикачен към писмото за гаранция, че наградата

отива при правилния човек. Такъв e-mail може да е измама с единствената цел да се събере информация за потребителя и организацията.

- *Web сайт измами* - често срещан тип Web сайт измама е измамата в структурата (frame-spoofing). Измамата в структурата използва съществуващите уязвимости в Web сайтовете. Например, ако нарушител иска информация за номера на кредитни карти. За да я получи, той вмъква фалшива структура в известен Web сайт за електронна търговия. Това е направено толкова добре, че няма и следа от нарушение на сайта. Клиентите, посещаващи сайта, без да знаят за намесата, предоставят номерата на кредитните си карти и други подробности по време на транзакцията. По този начин нарушителят получава необходимата информация. Друга използвана Web сайт измама работи чрез вграждане на нелегално съдържание, като порнографски добавки на привидно легитимни Web страници. Например извършителят може да стопанисва легитимен новинарски сайт, но да добави в страниците JavaScript съдържание, кое да отваря порнографски реклами. Когато потребителят потърси такива новинарски сайтове, вижда страницата в списъка с валидно име, но когато я отвори, се оказва незаконна. Друга подобна Web сайт измама е свързана със злепоставяне на Интернет доставчик (ISP). Потребителят получава e-mail от своя ISP със следното съобщение: "*Според нашите данни, заплащането за ваш акаунт за Интернет достъп е закъсняло. Може би сте го пресрочили? Важно е да се свържете с нас възможно най-скоро. За да подновите и формацията за акаунта си, моля посетете http://www.newpqrxyz.com*" Когато потребителят отвори хипервръзката, той го отвежда до нелегален сайт.

Заплахи за пълномощията

Заплахите за пълномощията (правата) са резултат от представянето на хакери като упълномощени потребители. Често срещана атака над правата е когато нарушителят открие мрежовата парола на потребител и влезе в системата. Нарушителите използват различни методи (като речникова атака или инструмент за разбиване на пароли, подробно описани по-долу) за разбиване (crack) на паролите.

Разбиване на пароли

Разбиването на пароли е занимание, предприето и от хакерите, и от системни администратори на организации. Системните администратори разбиват паролите по причини, свързани със сигурността, докато хакерите го правят, за да получат нелегален достъп. Системните администратори трябва да се подсигурят, че мрежовите пароли не са лесни за дешифриране, за да поддържат сигурността. Затова те често хакват пароли, за да окачествят устойчивостта на паролите.

За да разберете техниките за разбиване на пароли, трябва да знаете концепция на хеширане. Когато се създава акаунт, потребителят получава име и парола и подробностите се съхраняват в база данни. Но информацията за паролата не се съхранява в оригиналния си вид. Вместо това паролата е криптирана и съхранена под формата на хеш код {hash).

Хешът представлява криптиращ алгоритъм. Обикновено в базите данни хешированието се състои от добавяне или заместване на произволен брой символи към оригиналната серия символи. Представете си например, че паролата, която използвате е Maria и в полето за адрес сте въвели номер на къщата 497. Хешираната парола може да е криптирана като 45M64a8ril2a.

Така или иначе, същинската криптираща хеш функция приема вход с всякаква дължина и генерира изход, известен като хеш стойност, с фиксирана дължина. Знаем функцията, а е невъзможно да изчислим оригиналния вход от хеш стойността. Например, ако имаме хеш функция *F* и въведем вход x резултатът h, сметнат като *F(x) = h*, е еднопосочна трансформация. Това означава, че ако знаете функцията *F* и резултата h, не можете да изчислите x от *F* и h.

За да разбият пароли или хеш кодове, нарушителите използват инструменти за разбиване на пароли. Повечето инструменти работят, като криптират хиляди пароли в хеш кодове. След това сравняват резултата с хеш кодовете, съхранявани в базата данни. По време на сравнението инструментите използват определена логика или опитват всички възможни комбинации, докато не разбият всички пароли.

Друга техника, която нарушителите използват, за да разбиват пароли, е речниковата атака (dictionary attack). Първоначално терминът речникова атака се ограничаваше до. Ако паролата на атакуваната система съвпадне с

дума от речника, нарушителят влиза в системата. Речниковите атаки имат успех, когато паролите са обикновени думи.

Следват някои често срещани речникови атаки:

- Кратки атаки

За разлика от нормалните речникови атаки, където серия пароли се изпробват върху една система, в кратките атаки малко на брой пароли се изпробват върху няколко системи.

- Атаки “Груба сила” (brute force attacks)

Това е усложнена версия на традиционната речникова атака, практикувана от вътрешни нарушители, с цел допълнителни привилегии и разрешения в мрежата. При нея нарушителят първо се сдобива със списък от използвани пароли от истински източник. После хешира тези пароли, използвайки криптиращите схеми на атакуваната система. След това сравнява стойностите на резултатния хеш код и хешираните пароли, съхранявани в базата данни. Сравнението е успешно, когато хеш стойността от речника съвпада с хеш стойността от базата данни.

Заплахи за приложенията

В ранните дни на Интернет, Web сайтовете показваха само статични страници. Промяната на кода на тези сайтове беше лесна, защото можеше да се види по-голямата част от него. Все пак не бе възможно да се изложи на риск сигурността на приложенията и платформите, върху които се намираха тези сайтове. Самите мрежи, сигурността на платформите и физическата сигурност бяха достатъчни, за да защитят тези приложения от всяка заплаха.

Днес много Web сайтове се използват за електронна търговия и са динамични. Повечето от тях се състоят от приложения, които могат да взаимодействат по между си, както и със сървърите за бази данни и с други компютри. Тези приложения съдържат и обекти, които лесно могат да се изложат на риск. Има няколко механизма за сигурност - *защитни стени (firewalls)*, които позволяват достъп на външния свят само до определени портове, и *системи за вътрешна защита на операционни системи и мрежи*, които защитават от повечето уязвимости в глобалната мрежа. Но понякога

може да се случи така, че хакерът да пробие системата точно през порт, отворен за Web сайт.

Някои често срещани заплахи за приложенията:

- Превишаване на правата;
- Уязвимости на разработен от независим производител или персонализиран софтуер;
- Отказ на услуги (DoS);
- Разпределен отказ на услуги (DDoS);
- DNS превземане;
- Уязвимости на Web сървърите;
- Превземане на сесии;
- Атаки по e-mail;

Превишаване на правата

Превишаването на права е налице, когато нарушителят има възможност да повиши правата и привилегиите на неговата система до ниво, по-високо от нивото, на което трябва да бъдат. Ако такава атака успее, нарушителят може да добие привилегии и достъп до нива, високи колкото главната директория в UNIX система или административните права на машина с Windows.

Уязвимости на независим и персонализиран софтуер

Доставчиците на услуги често използват софтуер, разработен от трети лица (third-party) и го преработват (персонализират) в зависимост от нуждите си. Все пак персонализираният или разработеният от трети лица (независимият) софтуер съдържа някои вратички, предлагащи възможности за проникване в системите на една мрежа. Поради тази причина е съществено да се създават, използват и поддържат сигурни HTTP приложения.

Уязвимости на Web сървърите

Много от примерите за уязвимости, открити в Web сървърите, съществуват, защото сървърите предлагат повече от една услуга едновременно, дори и тези услуги да не са необходими.

От гледна точка на сигурността сървърът трябва да предлага само една услуга едновременно. Например, ако основната цел на сървъра е да съхранява (хоства) Web сайтове, трябва да бъдат спрени всички останали услуги, които не се използват. Ако не бъдат спрени, тези услуги представляват риск за сигурността, тъй като атакуващият може да използва някоя от тях, за да предприеме атаката си.

Друга открита вратичка в Web сървърите е, че те позволяват на нарушителите да заобикалят ограниченията на достъпа за файлове с дълги имена. Освен това някои Web сървъри предлагат достъп до нивото на свръхпотребителя (root), което може да представлява важна заплаха. Достъпът до нивото на свръхпотребителя е подобен на достъп до административно ниво. Ако администратор не ограничи правата до различните нива и даде административни права на всички потребители (което между другото е причина за сериозни опасения за сигурността), това би означавало всички потребители да имат неограничен достъп до информацията на сървъра.

Web сървърите също така са изправени и пред заплахата от лошо написани CGI скриптове. CGI е скриптов език от страна на сървъра, който позволява на Web страниците да си взаимодействат и да са динамични. Повечето сървъри използва CGI като скриптов език от страна на сървъра по подразбиране. CGI сам по себе си е сигурен скриптов език. Все пак, един лошо написан CGI скрипт има определени вратички, заради които Web сървърите стават уязвими. Например CGI скриптовете влияят на свойства на Web сървърите и ако не са добре написани, това привлича вниманието на нарушители. Лошо написаните CGI скриптове дава шанс на атакуващите да проверят възможностите на сървъра и така да го манипулират.

Превземане на сесии

HTTP е несвързан протокол. Това означава, че при приключване на първоначалния обмен на съобщения между клиента и сървъра връзката прекъсва. С други думи, HTTP протоколът не запазва състоянието на сесията.

Тъй като HTTP е несвързан по подразбиране, идентификаторът на сесиите (ID) е един вид заобикаляне на проблема със запазването на състоянието на сесията. Cookies осъществяват идентичността на сесиите. В глобалната мрежа на всяка комуникационна сесия между клиент и сървър

се задава идентичност, част от процеса на автентикация, която пази информация за активността на сесия и за връзката с потребителя. Връзката се поддържа чрез задаване на номер за автентикация на потребителя. Идентификаторът на сесията и номерът за автентикация на потребителя се съхраняват в база данни с транзакции на сървъра.

При последващ достъп до сайта идентичността на сесията на потребителя и информацията за автентикация пътуват заедно с URL адреса и същинското съдържание на сайта. Цялата тази информация е групово позната като разширен URL адрес (или просто URL). При превземането на сесии нарушителят превзема идентичността на сесията чрез достъп до данни на сървъри и мрежи. Нарушителят променя детайлите на разширения URL, използвайки текстов редактор, и след това се свързва отново с Web сайта с разширения URL.

Атаки по e-mail

Атакуващите експлоатират уязвимостите на електронната поща, за да разруша мрежови услуги и системи в глобалната мрежа. Следват няколко e-mail атаки използвани от нарушители в глобалната мрежа:

- e-mail бомбардиране;
- изпращане на спам (нежелани) съобщения;
- e-mail подслушване и измами.

E-mail бомбардиране

E-mail бомбардирането е метод, при който нарушителят изпраща на мишената идентични писма едно след друго, като по този начин препълва кутията му.

E-mail бомбардирането е ценно от гледна точка на ограничение на достъпа и злоупотреба с мрежови ресурси. За хора, използващи e-mail акаунти, които предоставят определено пространство, e-mail бомбардирането може да доведе до пропускане на важен e-mail поради препълване на пощенската им кутия. С други думи, e-mail бомбардирането може да срине e-mail сървър, с което да причини пропускане или забавяне на легитимен e-mail. Освен това то може да блокира мрежови ресурси, което да повлияе на други системи от същата мрежа.

Пример за такъв инструмент - Mail Bomber.

Изпращане на спам съобщения

Изпращането на спам съобщения (нежелан e-mail) е метод, при който атакуващите изпращат e-mail на стотици или хиляди потребители. При тази атака атакуващият се записва в пощенски списък или получава списък с адреси от компании, които ги поддържат. След това той изпраща нежелан e-mail на тези потребители продължително време.

Изпращането на спам съобщения не може да бъде реално контролирано, защото всеки, който има валиден e-mail адрес може да изпраща съобщения до всеки друг e-mail адрес, нюзгрупа (news group) или електронен бюлетин (BBS, Bulletin Board System). Заради него до или през Web сайт могат да се изпращат голямо количество електронни съобщения. Това може да се превърне в DoS атака, тъй като предаването на прекадено много e-mail съобщения създава голям трафик в мрежата, забавя e-mail сървъра и използва всички свободни ресурси на системата.

E-mail подслушване и измами

Друг вид атака е e-mail подслушването (sniffing) и измамата (spoofing). При e-mail подслушването атакуващите прихващат e-mail съобщения по мрежата, преди те да са пристигнали при получателя си.

За да вземат информацията, атакуващите използват софтуер за подслушване на пакети. Тези приложения могат да прихващат информация, пътуваща от един компютър към друг. Веднъж прихванали e-mail съобщение, атакуващите получават тайна информация за номера на кредитни карти, потребителски имена и пароли или важна информация за организации:

При e-mail измамата атакуващият прихваща e-mail съобщение по мрежата и променя информацията в него със злонамерена цел. При някои случаи на измами атакуващите главно осъществяват пренасочване на прихванати съобщения под променена самоличност.

Заплахи за поверителността

Заплахите за поверителността включват различни атаки с подслушване. Някои от тях са:

- мрежово подслушване;

- подслушване на радио сигнал

Мрежово подслушване

Мрежовото подслушване включва наблюдаване на данните, които се предават по локални мрежи, и извличане на желаната информация. За интерфейс контролер служи мрежово устройство, което наблюдава данните в мрежата. Често се използва от хакери за улавяне на данни за подслушване.

Мрежовото подслушване е често срещан метод за подслушване в мрежа. Той включва прихващане на пакетите от данни, пътуващи по мрежата, и дешифрирането им за извличане на критичната информация - потребителски имена и пароли. Заради съществуването на такива атаки се препоръчва предаването на данни в криптиран вид.

Подслушване на радио сигнал

Подслушването на радио сигнал включва прихващането на радио сигнал, издаван от компютри и мобилни телефони. Този тип атака не се среща често, тъй като изисква скъпо оборудване (като антени и настройващо оборудване) за улавяне данните по мрежата. В наши дни има софтуер, който позволява на компютри да се настройват по радио сигнали на компютри или мобилни телефони на всички вълни.

Заплахи за контрола на достъпа

Както подсказва името, заплахите за контрола на достъпа включват атаки, целящи достъп до системи в Интернет. Най-често срещаният метод е разбиване на пароли. Други атаки включват достъп до файлове с пароли или комуникационни точки като модеми или използване на софтуер, който дава достъп до вътрешни или външни системи през задна вратичка (backdoor), експлоатирайки този начин пробиви в мрежата.

Нарушителите обикновено използват backdoor атаки, за да получат достъп до системи. След като вече има достъп, нарушителят най-често оставя входна точка (също наричана backdoor) отворена чрез манипулиране на програмен код, така че да може да влезе в системата отново от това място. Нарушителят оставя вратичката отворена, така че системата остава отворена дори и ако присъствието му е било забелязано. Задните вратички помагат на нарушителите да избегнат проследяване чрез вписвания в регистрационните

файлове (logs). С други думи, влизането с backdoor заобикаля засичащия механизъм на системата.

Вируси

По дефиниция компютърният вирус е малка програма, която се копира с друга програми или файлове. При изпълнение на заразената програма вирусът също се изпълнява. Компютърните вируси са наречени така, защото споделят някои от характеристиките на биологичните вирус. Компютърният вирус се предава от компютър на компютър, също както биологичният се предава от човек на човек. Както биологичният вирус използва клетката, за да се възпроизвежда, компютърният се нуждае от изпълнима програма, за да си прави копия.

Вирусите може да не започнат да действат веднага след заразяването на системата. Те могат да стоят скрити известно време, докато не се активират на определена дата. Но дори и когато вирусът лежи скрит, системата е заразена.

Ефектът от вируса може да варира от просто показване на дразнещо съобщение (например вирусът WM97/Class-D, който постоянно показваше съобщения като *"I think 'username' is a big stupid jerk*") до изтриване на специфични файлове или форматиране на твърдия диск (например вирусът CIH, който се опитва да препрограмира Flash BIOS и може да причини сериозни вреди).

Най-често използваните носители, чрез които се разпространяват вирусите, могат да се класифицират в следните категории:

- Преносими носители
- Файлове, свалени от Интернет
- Файлове, прикачени към e-mail

Преносими носители

Най-често срещаните носители на вируси са преносимите устройствата за физическо съхранение на данни: компактдискове, преносими дискове, флаш памети и др. При използване на заразени носители, вирусът се пренася на твърдия диск.

Файлове, свалени от Интернет

Интернет е друга среда, чрез която вирусите се пренасят на компютрите поради голямата достъпност на freeware, shareware, софтуер, игри, MP3 файлове, помощни инструменти и снимки за сваляне. Повечето сайтове уверяват, че предлаганата от тях информация не е заразена. Все пак, винаги има шанс такъв материал да е заразен. При сваляне на заразени файлове от Интернет, компютър може да се зарази. На свой ред, при работа в мрежата и споделяне на зарани файлове, цялата мрежа може да се зарази.

Прикачени към e-mail файлове

E-mail съобщенията са друга среда за разпространение на вируси от Интернет. Заразен с вирус документ е изпратен по e-mail като прикачен файл. Когато получателят отвори документа, намиращият се в него вирус се активира. Следователно вирусът се самокопира като няколко изпълними копия, прикачва се към други електронни писма и в края на краищата изпраща заразени съобщения до всички пощенския списък или списъка с адресите на този потребител.

Например потребителят може да получи игра като прикачен файл от неизвестен подател. Когато зареди играта, намиращият се в нея вирус се изпълнява автоматично без знанието на потребителя. В този случай вирусът може не само да повреди компютърните файлове на този потребител, но и да изпрати съобщения прикачен файл-вирус до всички адреси от пощенския му списък. Потребите няма дори и да забележи, че съобщенията за изпратени от неговия акаунт. Получателят може да не забележи, че това е потенциално разрушителен e-mail, защото е изпратен от познат.

Най-вероятни мишени на атаки с вируси

В наши дни всеки с компютър и e-mail акаунт може да е мишена на вирусна атака. Все пак, най-вероятни мишени на такива атаки са професионалисти, работещ в IT организации, служители в организация (отделно от IT), използващи информационни технологии, за да изпълняват задълженията си, или лица, използваш информационни технологии, за да изпълняват рутинни операции. Причината е това, че тези хора по-често взаимодействат с Интернет в сравнение с хората, които използват компютрите си вкъщи.

Само че във всяка организация има служители на длъжности, по-уязвими към вирусни атаки от други. Например служители, имащи мобилен достъп до мрежата на организацията, са много вероятни жертви на тези атаки, тъй като използва клиентски мрежи, които могат да съдържат вируси. В този процес тези хора могат, без да знаят, да пренесат вирус в мрежата на организацията. Друга длъжност е тази на супервайзорите и ръководителите на екип, които може да имат достъп до компютрите на своите подчинени. Компютърът на някой подчинен може вече да е заразен с вируси и като резултат компютърът на началника или ръководителя на екип да се инфектира. Хора, които споделят компютри или разменят преносими носители, за да пренасят информация, също за изложени на такива атаки.

Типове вируси

В зависимост от поведението им и областта от компютъра, която атакуват, вирусите се класифицират в две категории: *стартово-секторни* и *файлови* вируси. Следващите вируси, на свой ред, могат да се подкласифицират пак на стартово-секторни и файлови вируси:

- Макро вируси;
- Скриптови вируси;
- Паразитни вируси;
- Потайни вируси;
- Полиморфни вируси;

Стартово-секторни вируси

Стартово-секторните вируси се разпространяват основно, когато системата се стартира от заразен носител. Вирусът се прочита от заразения стартов сектор на носителя и се записва в стартовия сектор (например master boot record, MBR) на системния диск. MBR е първият сектор, от който системата чете, когато започне процесът на зареждане. Сега, когато и да се зареди системата, вирусът се зарежда в системната памет. Той заменя и променя файлове на операционната система в стартовия сектор. Пример за стартово-секторни вируси са Disk Killer, Stoned и Michelangelo.

Файлови или програмни вируси

Файловите вируси, още наречени програмни вируси, заразяват изпълними файлове с разширения .EXE, .COM и .DLL (въпреки че в някои

случай инфектират също и изпълними файлове със .SYS, .DRV, .BIN, .OVL и други по-рядко срещани разширения).

Вирусите, които припокриват или променят съдържанието на заразените файлове, са лесни за откриване, защото изменят размера на файла, към който се закачват. Файловите вируси могат да се разпространяват чрез преносими носители или Интернет. Примери за файлови вируси са Dark Avenger и Cascade.

Има няколко файлови вируса, които променят и разширението на изпълнимия файл. Например те могат да променят оригиналното разширение на файла .COM на .EXE и след това да създадат нов файл със същото име и разширен .COM. Трудно е да се засече подобен заразен файл, тъй като в този случай размерът на новосъздадения от вируса файл е същият като на оригиналния.

Най-успешните файлови вируси са резидентни (постоянно присъстващи в паметта) вируси, които се зареждат в паметта първия път, когато стартирате заразен файл, и след това поемат пълния контрол върху компютъра. Такива вируси инфектират главно допълнителни програми всеки път, когато заразеният файл изпълни или дори когато се правят нови директории. Все пак има много нерезидентни вируси, които просто заразяват един или повече файлове при всяко стартиране на заразения файл. В традиционните .EXE и .COM файлови вируси нерезидентните вируси не са много успешни поради неспособността си за разпространяване на инфекцията на световно ниво.

Макро вируси

Макро вирусите са основно инструкции за приложения, написани на езици като WordBasic или Visual Basic. Тези вируси преживяват главно в документни файлове на приложения като например Word документи, Excel таблици или файлове - шаблони на същите тези приложения. Първоначално документите се разглеждаха просто като файлове за съхранение на данни и никой не е очаквал да бъдат заразени с вируси. С появата на макро вирусите обаче, всяко приложение, поддържащо макроси, които са прикрепени (или могат да се включат по някакъв начин) в документен файл, има възможността да се зарази с макро вирус.

Не е задължително макросите да са прикрепени към документни файлове, за да се зарази приложението, което ги съдържа. Например в

началото на 1999 беше открит първият вирус, наречен CSC/CSV.A, за приложения, поддържащи макро езика CorelScript. Все пак, CorelScript е език, базиран на няколко файла с код за макроси. Поради тази причина вирусът не беше голяма заплаха за потребителите, които използваха други приложения, поддържащи CorelScript. Това е случай на съвместимост на приложения, от който се заключава, че не можете да напишете един макро код, който да работи на всички приложения, поддържащи CorelScript. В случая на CSC/CSV.A, поради последици от съвместимостта с CorelScript, вирусът не успя да се самокопира достатъчно бързо, тъй като беше съвместим само с един тип приложение, за което беше създаден.

Макро вирусите представляват огромна заплаха в сравнение с традиционни стартово-секторни или файлови вируси. Степента, до която макро вирусите влияят на системата, е много висока, защото те са лесни за създаване. Все пак макро вирусите не могат да въздействат на цялата система, защото са специфични отделни приложения. Макро вирус, който повреждa Word документи, няма повлияе на документи, създадени с Lotus Notes или Star Office. Макро вируси са по-популярни, защото е лесно да бъде инфектирано приложение, което поддържа макроси. Друга причина, която прави макро вирусите доминиращи, е широката употреба на макро езика VBA (Visual Basic for Applications). VBA е мощен и лесен за научаване език и затова е много известен между професионални програмисти и хакерите. Освен това той дава достъп до сложни Windows API програми. Достъпът до тях е труден с езиците от ниско ниво. Това е причина много хора, които не са създавали вируси преди заради сложността на езиците ниско ниво, сега да могат лесно да пишат вируси.

Скриптови вируси

Скриптовите вируси заразяват приложения, написани на скриптови езици като например JavaScript (JS) и Visual Basic Script (VBS). JS и VBS езиците се поддържат от Windows Scripting Host (WSH), който е активен скриптов компонент, предоставен от операционните системи Windows.

Скриптовите вируси се разпространяват широко, тъй като много компютри ползват WSH, за да стартират приложения, базирани на скриптове.

Друга причина за широкото разпространение на скриптовите вируси е, че писането на код за тях е лесно. JS и VBS, които са скриптови езици от

високо ниво могат да се разберат толкова лесно, че дори някой програмист без подготовка може да напише скрипт на някой от тези езици. Не се нуждаете от големи познания по програмиране, за да напишете код за скриптов вирус. Например операторите за обработка на низове решават в JS и VBS скриптовете много проблеми, които бяха почти невъзможни в езиците от ниско ниво.

Като макро вирусите, скриптовите вируси могат да се разпространяват чрез обмяна на преносими носители, при достъп до e-mail или при споделяне на директории по мрежата. Скриптовите вируси могат също така да се разпространяват чрез ActiveX приложения. ActiveX е приложение на Microsoft, което позволява на Web страниците да свалят компоненти. Това означава, че при достъп до Интернет, чрез ActiveX могат на машината да се свалят модули, инфектирани със скрипт вируси. Освен това скриптовите вируси могат да се разпространяват чрез приложения като файлове за Windows Help, инсталационни файлове на Windows - файлове от регистрите на Windows (.REG файлове). За разлика от макро вирусите, скриптовите вируси могат да се разпространяват и чрез Internet Relay Chat (IRC) - система, която позволява онлайн разговори по Интернет.

Паразитни вируси

Паразитните вируси заразяват изпълними файлове, наричани още програми. Когато е стартирана заразена изпълнима програма, първо се активира паразитният вирус. Все пак, за да скрие присъствието си, вирусът стартира оригиналната програма, към която е прикачен. Докато операционната система разбере, че в програмата има вирус, той получава същите права като тези на програмата. Това на свой ред позволява на вируса да се самокопира, да се инсталира в паметта и да унищожава файлове и други приложения. Jerusalem е добре известен паразитен вирус, който забавя компютъра и изтрива всяка програма, която потребителят се опита да стартира.

Потайни вируси

Потайните вируси са вид стартово-секторни вируси, които заразяват файловете и началния (boot) запис на диска. Тези вируси наблюдават системните функции използвани от програмите, за да четат файлове или физически блокове от диска и променят резултатите от тези функции, така че програмите, които се опитват да четат от тези области, виждат

оригиналната незаразена форма на файловете вместо действителната инфектирана форма. Затова, щетите или промените, причинени от вируса, не се откриват от антивирусните програми. Все пак, ако искате да засечете вируса с антивирусна програма, той трябва да се намира в паметта.

Полиморфни вируси

Полиморфните вируси създават различни (все още напълно операционни) свои копия с надеждата, че скенерите за вируси няма да могат да засекат всички негови копия. Тези вируси се маскират и променят вида си с всяко заразяване. Някои полиморфни вируси са известни и като криптирани вируси, защото използват техниката на криптирането, за да се скрият от антивирусния софтуер. Тези вируси криптират главния си код и използват случайно избран набор от команди, за да декриптират кода.

Полиморфните вируси са трудни за откриване, затова трябва да се използва антивирусен софтуер, който сканира по алгоритъм. Сканирането по алгоритъм е по-сложно и усъвършенствано от нормалното сканиране, базирано на низове, което се използва за засичане на прости вируси. Пример за полиморфен вирус е MtE, което означава Mutation Engine (Двигател на Мутации).

Предотвратяване, засичане и възстановяване от атаки с вируси

Днес заплахите от вируси са познати като уязвимост от Интернет. За да се предотврати заразяването на компютри по целия свят, има няколко мерки за предотвратяване, засичане и възстановяване, които една организация трябва да изпълни.

Мерки за сигурност за предотвратяване на атака от вирус

Най-добрият начин за отделни хора и мрежови администратори на организации да предпазят мрежата и отделни компютри от вируси е да се вземат следните антивирусни мерки:

- На първо място, всеки, който се занимава с компютри, трябва да има основни познания за различните видове вируси и начина им на функциониране. Това съзнание ще помогне на хората да избегнат основните заплахи от вируси. Например Word или Excel

документите съдържат макроси и затова са най-предразположени към атаки от вируси.

- Избягване на стартиране на макроси в документи, ако не е известна функционалността им.
- Друга мярка за сигурност, която може да намали последиците от инфектиране с вирус на компютъра, е правенето редовно на backup файлове (копия на програми или файлове, съхранявани отделно от оригиналите) на твърдия диск.
- Инсталиране на антивирусна програма и редовното и подновяване.
- Мрежите трябва да са изградени така, че само упълномощени потребители да имат достъп до мрежовите ресурси. За да осъществите това, съществуват различни инструменти, които предпазват от нелегален достъп.
- Организациите трябва да осигурят специални машини, чиято единствена цел да е тестването на нов софтуер, файлове или дискове. Файловете, които трябва да се споделят по мрежата, трябва да се сканират на тези машини, преди някой да получи достъп до тях.
- Преносът на изпълними файлове от и до външни източници трябва да се блокира. Блокирането на трансфера на изпълними файлове по този начин ще подсигури и поддържа цялостната защита срещу атаки от вируси.

Засичане на вирусни атаки

За да се улеснят засичането на вируси в мрежата, организациите трябва да следят мрежата за признаци на необикновено държание или дейност. Заразените с вируси компютри имат няколко често срещани симптома:

- Мрежата се забавя и данните често се блокират.
- Компютърът може да работи по-бавно или дори да спре да действа.
- Може да има необикновен брой програмни грешки или грешки в паметта.
- Размерите и последно променените дати на обекти, които са заразен може да са променени.

- Може да има необяснима загуба или модификация на информация.
- Твърдият диск на компютър може да изглежда пълен дори и да има свободно място.
- Командата CHKDSK не показва верен брой байтове.
- Отварянето или стартирането на приложения може да отнема повече време от обикновено.
- Може внезапно да се появят непознати типове файлове в системата.
- Документите може да показват странни символи, въпреки че са записани правилно.
- Клавиатурата може да издава звуци дори и когато не реагира.
- Може мистериозно да се стартират програми под видимо защитен потребителски профил.

Възстановяване от атаки с вируси

Персоналните компютри може да не действат по очаквания начин поради редица причини. В повечето случаи причината за неочакваното поведение не е атака от вирус, а по-скоро обикновен мрежов проблем. Все пак, повечето офис потребители не знаят това. Служителите от техническата поддръжка трябва да обучат офис потребителите как да предотвратят или поправят такива малки проблеми. Например, ако потребител подозира, че компютърът му се държи необикновено той трябва да се върже с координатора на техническата поддръжка в организацията. Освен това той трябва да улесни процеса по решаване на проблема, като води бележки по проблема, съобщенията за грешки, които може би се появяват задачата, която се изпълнява в момента на грешката, историята на проблема и всякакви промени, направени в хардуера или софтуера на системата, наличен от греди появата на проблема.

След внимателен оглед на проблема, ако техническата поддръжка подозира, че компютърът е заразен с вирус, компютърът трябва да се изолира от мрежата. На първо място трябва да се помисли за възстановяване и създаване на backup файлове на данните на заразения компютър. Като част от това упражнение, компютърът трябва да се провери чрез използване на софтуер, сканиращ за вируси. Освен това трябва да се подсигури, че вирусът е премахнат от всяка част на мрежата, защото има много скрити вируси,

които остават незасечени от антивирусните програми и могат да се активират и разпространят по-късно.

За да подсигури, че компютърът е успешно възстановен от вирусна атака, системният администратор трябва да изпълни стъпки като:

- Да оцени размера на причинените щети. Ако компютърът е част от мрежа, техническата поддръжка трябва да открие колко компютъра са повлияни и дали има други места, които са засегнати. Заразените компютри трябва да се изолират от мрежата, така че вирусът да не се разпространява. За да предотвратите следващи атаки, трябва да идентифицирате източника на вируса, което може да стане с преглеждане на регистрационните файлове на компютрите и сървърите.
- Да провери backup сървърите за заразяване с вирус. След премахване на заразените компютри от мрежата трябва да се направи проверка на backup сървърите за заразяване с вирус. Необходима е проверка с антивирусни програми. Добре е да се направи копие на данните, съхранявани на сървъра, за да можете да се възстановят, ако се загубят данни по време на изчистването.
- Да провери за вируси всички компютри в мрежата. След това с антивирусни програми се установява кои данни и програми са заразени.

Лекция 3

Видове мрежи, заплахи и уязвимости в TCP/IP мрежите

В предишния лекционен курс по мрежово администриране бяха представени основни мрежови топологии и класификации по различни признаци. Ще припомня две известни класификации, които включват технологията на пренос и площта, която мрежата покрива. Общо казано това са:

- Разпределени (broadcast) мрежи;
- Мрежи “от точка до точка”.

Разпределени мрежи

В разпределените (broadcast) мрежи всички компютри споделят един комуникационен канал. Ако един компютър иска да комуникира с друг, се изпраща съобщение (под формата на пакети) до всички компютри от мрежата. Пакетите съдържат адреса на компютъра, до който трябва да се изпрати съобщението. Когато някой компютър получи пакет, той проверява прикачения към пакета адрес. Ако адресът от пакета съвпада с адреса на компютъра, той обработва пакета. Ако пакетът обаче не е предназначен за този компютър, той просто бива игнориран.

В една разпределена мрежа пакетът може да бъде изпратен и до всички компютри в мрежата или да се достави до точно определена група компютри. Доставянето на пакетите до всички компютри в мрежата се нарича разпределяне (broadcasting), а доставянето им до определена група компютри в мрежата се нарича множествено разпределяне (multicasting).

Разпределената технология се осъществява основно в малките мрежи, в които има малко компютри и които се намират на малка географска област.

Мрежи “от точка до точка”

За разлика от разпределените мрежи, мрежите “от точка до точка” се състоят от няколко комуникационни канала между компютрите в една мрежа. Пакетите от данни може да минават по различни маршрути с различна дължина, за да достигнат до получаващия компютър (получателя).

Използва се рутиращ алгоритъм, за да се определи пътят на един пакет до получателя му.

Комуникационната технология “от точка до точка” се осъществява основно в големи мрежи, в които може да има подмрежи.

Заплахи и уязвимости на TCP/IP мрежите

TCP/IP е повдигнал много въпроси за сигурността, защото първоначално не е замислен да помни различните аспекти на сигурността. По-точно той е създаден, за да поддържа приложения и да осигури свързване и възможност за взаимодействие между мрежи. Освен това, контролирането на мрежи в такъв голям обем е трудно. През последните години слабостите на групата от TCP/IP протоколи отвориха големи дупки в сигурността, правейки мрежите уязвими за различни заплахи.

Начин на функциониране на TCP/IP

TCP е протокол, осигуряващ надеждно предаване на сегменти между съответните процеси на двата комуникиращи хоста. Той назначава пореден номер за всеки предаден байт и очаква потвърждение от приемащия хост за получените байтове. Това забавя доставката на данни за сметка на тяхната сигурност, която се характеризира със следните действия:

1) Създаване на логическо съединение (сесия) между двата комуникационни процеса (Session setup) – този тип съединение се изгражда между портовете на двата комуникиращи хоста. Използването на портове позволява на TCP да поддържа множество логически съединения, едновременно, между два хоста или между хост и множество хостове. Този процес се нарича мултиплексиране.

При изграждането на сесия се използва подход, наречен трикратно ръкостискане (three-way handshake). При този механизъм, целевият хост получава SYN сегмент от хоста-източник, на който отговаря с ACK/SYN сегмент и очаква неговото ACK потвърждение от инициализиращия хост. SYN сегментите дават възможност на комуникиращите хостове да установят стартови параметри за комуникацията – размер на прозореца (буфер), първоначален пореден номер (специфичен за всеки хост), максимален размер на сегмент. На фигура 1 е илюстрирано трикратното ръкостискане.

фигура 1 *Трикратно ръкостискане*

- Хост А изпраща SYN сегмент с включен начален пореден номер SEQ=5 (Sequence Number);

- Хост В потвърждава с ACK (5+1=6), включен бит SYN и началния си пореден номер SEQ=55;

- Хост А потвърждава ACK/SYN сегмента с ACK (55+1=56);

- Извършва се трансфер на данни след установяване на връзката. TCP използва пореден номер за всеки байт прехвърлени данни, които трябва да бъдат потвърдени от приемащия хост. Полето SEQ идентифицира първия байт от сегмента, а полето ACK съдържа очаквания следващ пореден номер като потвърждава всички данни до него. Например, ако Хост А предаде 3 сегмента с по 15 байта данни и началният пореден номер е 30, то следващият очакван пореден номер, заложен в ACK пакета, ще бъде 30+3.15=75;

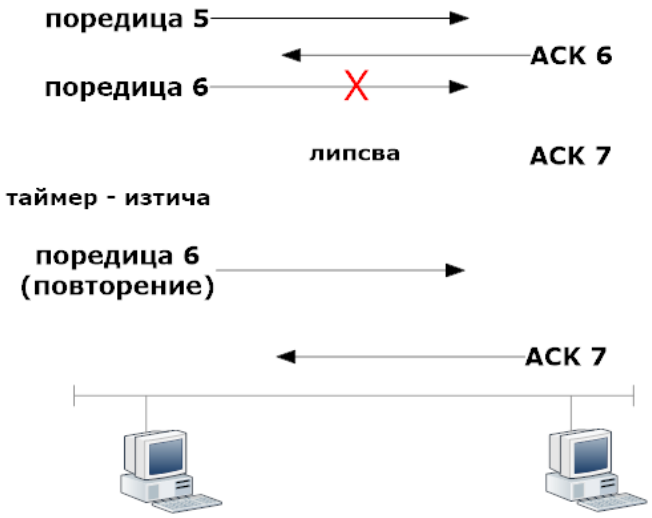
2) Контрол на последователността на данните в сегмента (Sequencing);

3) Контрол на потока от данни (Flow control) – гарантира, че входящият трафик няма да запълни буферите на приемащия хост и той ще може да обработи потока от данни, както и да отговори на запитвания от страна на предаващия хост. Механизмът, който се използва е познатият плъзгащ се прозорец (sliding window). Всеки хост поддържа такъв прозорец и контролира размера му спрямо моментните си възможности. Чрез него приемащият хост указва количеството данни, което може да буферира. Максималната стойност е 65535 байта и се определя от размера на полето, което е 16-битово (Window). Размерът на прозореца, потвържденията и поредните номера са байтово базирани, а не сегментно;

4) Поддържане на връзката при липса на данни за предаване – използва се служебно съобщение (keepalive) за поддържане на връзката. То не съдържа данни от по-горен слой, което означава, че полето за дължина е 0 и

следващият ACK номер не се увеличава. Ако такива връзки не се затворят, се увеличава натоварването на мрежата, особено в случаите, когато са много на брой;

5) Потвърждение на правилно приетите данни (Acknowledgement)– потвърждението е механизъм, позволяващ на хостовете да определят кога има загуба на данни. Приемащият хост не изпраща ACK потвърждение за изгубен пакет. При неполучаване на потвърждение за определен период от време (използва се таймер), изпращащият хост повтаря предаването на непотвърдените данни, които се съхраняват в т.нар. TCB буфер (блок за контрол на предаването) (фигура 2);



фигура 2 *Препредаване на липсваща поредица*

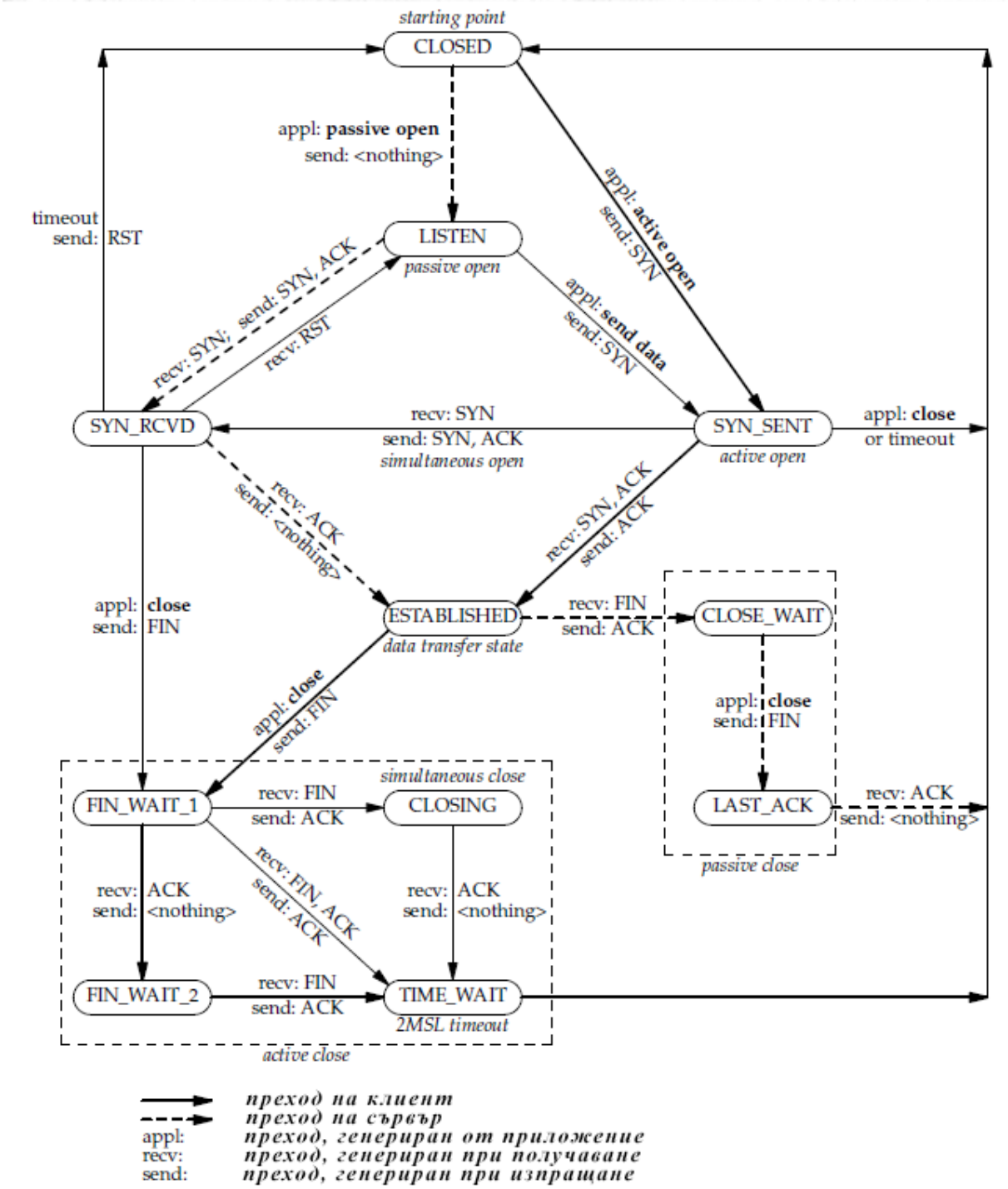
6) Разпадане на логическото съединение (Session teardown) – процесът е подобен на този за установяване на сесия. Използва се трикратно ръкостискане, където SYN сегментът е заменен с FIN. Затварящият хост изпраща FIN сегмент към другата страна за затваряне на сесията. Приемащият хост потвърждава с ACK/FIN отговор, които включва и собствен FIN, очакващ потвърждението му от хоста-инициатор. След тази процедура сесията се затваря. Състоянията, през които преминават двете страни, използващи TCP съединението са:

- за клиента – CLOSED, SYN-SENT, ESTABLISHED, FIN-WAIT-1, CLOSE-WAIT, FIN-WAIT-2, CLOSING, LAST-ACK, CLOSED;
- за сървъра – CLOSED, LISTEN, SYN-RESEIVED, ESTABLISHED, FIN-WAIT-1, CLOSE-WAIT, FIN-WAIT-2, CLOSING, TIME-WAIT, CLOSED.

означение	състояние
LISTEN	Състояние на сървъра, очакващ създаване на съединение
SYN-SENT	Състояние на клиента, изчакващ отговор на SYN заявката си за създаване на съединение
SYN-RECEIVED	Състояние на сървъра, очакващ потвърждение ASK на изпратения от него SYN
ESTABLISHED	Състояние и на двете страни при изградено и използващо се съединение
При закриване на съединение инициатори могат да бъдат и двете страни	
FIN-WAIT-1	Състояние на инициатора за закриване на съединението след изпращане на заявката FIN. Очаква потвърждение от другата страна
CLOSE-WAIT	Състояние на отсрещната страна след отговор ACK на заявката за закриване (FIN)
FIN-WAIT-2	Състояние на инициатора след получаване на ACK потвърждението от другата страна на неговата FIN заявка
LAST-ACK	Състояние на отсрещната страна след изпращане на FIN отговор
CLOSING	Състояние на двете страни при едновременно затваряне на използваното съединение
TIME-WAIT	Състояние на изчакване за определено време преди закриване на съединението

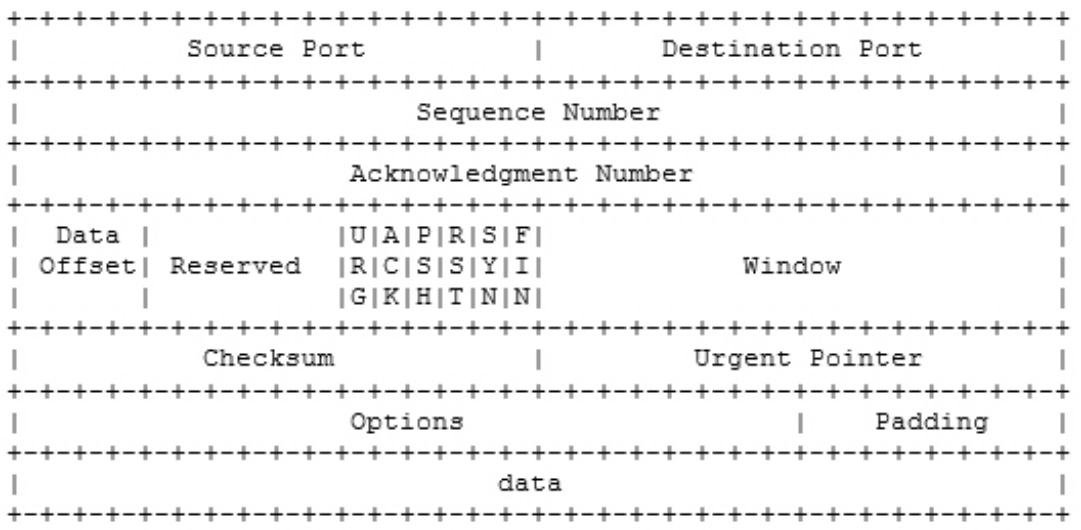
таблица 1 *Състояние на двете страни на TCP съединение*

Състоянията и събитията, предизвикващи тяхната промяна са представени на фигура 3.



фигура 3 *TCP състояния*

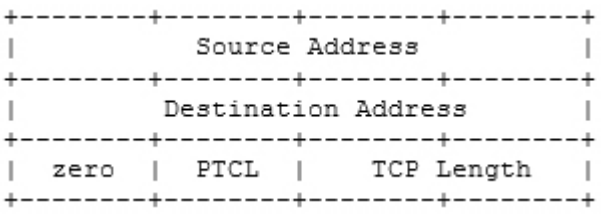
Структурата на TCP сегмента [RFC 793] е показана на фигура 4.



фигура 4 *Структура на TCP сегмент*

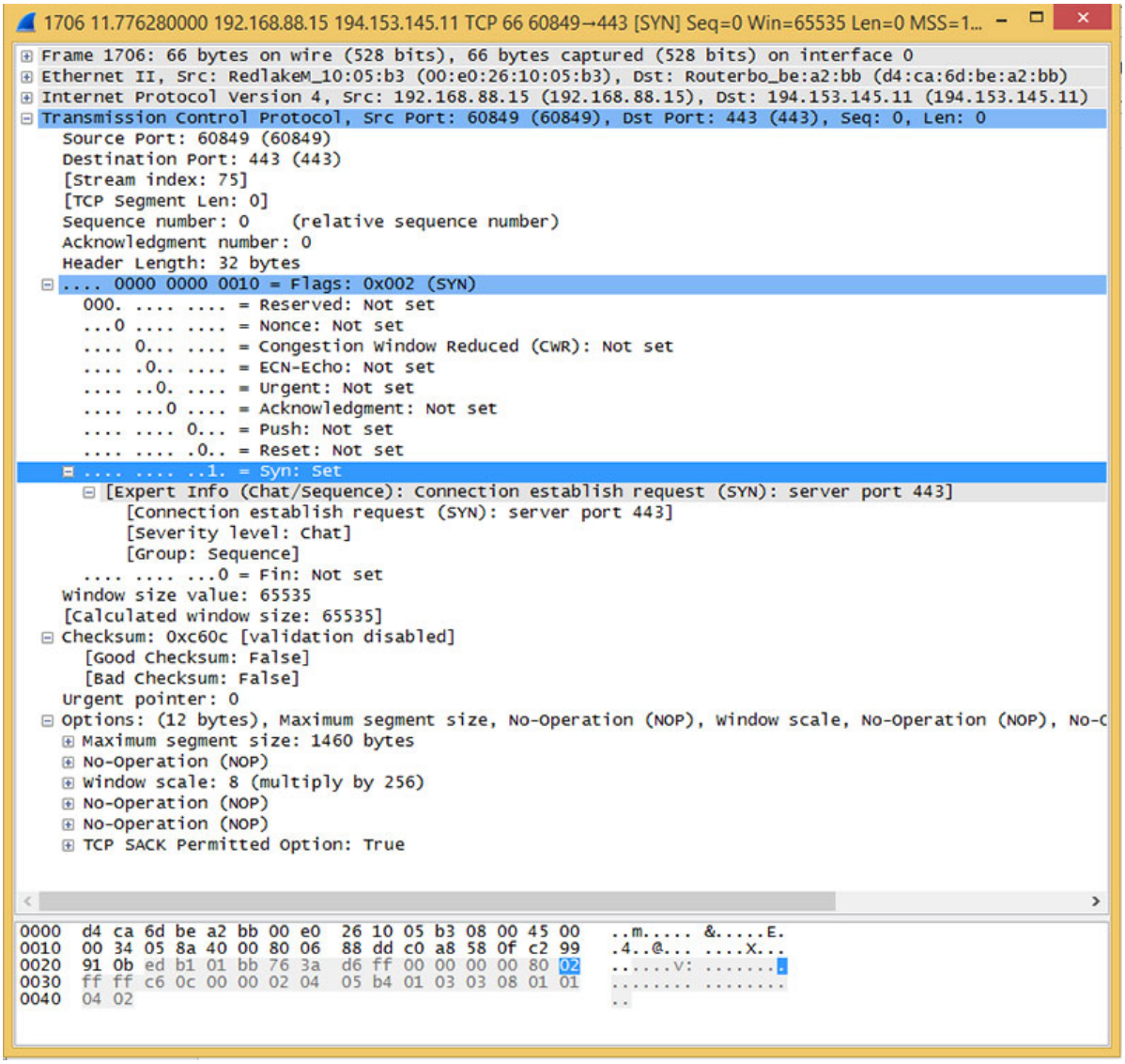
Source Port – 2-байтово поле, идентифициращо изходния порт (сокета) на комуникационния процес на хоста-изпращач;
Destination Port – 2-байтово поле, идентифициращо входния порт (сокета) на комуникационния процес на хоста-приемник;
Sequence Number - 4-байтово поле, съдържащо поредния номер на първия октет от данни в сегмента;
Acknowledgment Number - 4-байтово поле, съдържащо поредния номер на началния октет на следващата последователност от данни;
Data Offset – 4-битово поле, указващо началото на данните, следващи TCP хедъра. Налага се поради променливата дължина на хедъра.
Reserved – 6-битово поле е резервирано и винаги е запълнено с 0.
URG (Urgent) – 1-битово поле, задаващо висок приоритет на данните. Активира Urgent Pointer указателя, сочещ първия байт от сегмента след спешните данни;
ACK (Acknowledgment) - 1-битово поле, задаващо сегмента като потвърждение;
PSH (Push) - 1-битово поле, което при стойност 1 задължава приемащия хост да не задържа пристигащите данни, а да ги изпрати към приложния процес от по-горен слой;
RST (Reset) - 1-битово поле, задаващо стойност 1, ако е необходимо прекъсване на сесията;
SYN (Synchronization) - 1-битово поле, задаващо инициализирането на сесия;
FIN (Finish) - 1-битово поле, задаващо финализирането на сесия от изпращащия хост;
Window - 2-байтово поле, задаващо свободния размер на буфера (в байтове) на приемащия хост. Стойността варира според възможностите на хоста. Ако стойността на прозореца е 0, то този хост не може да приема данни в момента.
Checksum - 2-байтово поле, съдържащо контролната сума, изчислена върху TCP хедъра, IP псевдохедъра (фигура 5) и данните;
Urgent Pointer - 2-байтово поле, задаващо байта в сегмента, откъдето започват не-спешните данни. Включва се при URG=1;
Options – поле с променлива дължина. Включва опции, избрани от изпращащия хост (например опцията за максимален размер на сегмент - MSS).

Псевдохедърът спомага за откриването на погрешно насочени сегменти. Съхранява се в TCB (transmission control block) буфера и включва показаните на фигура 5 полета от IP хедъра.

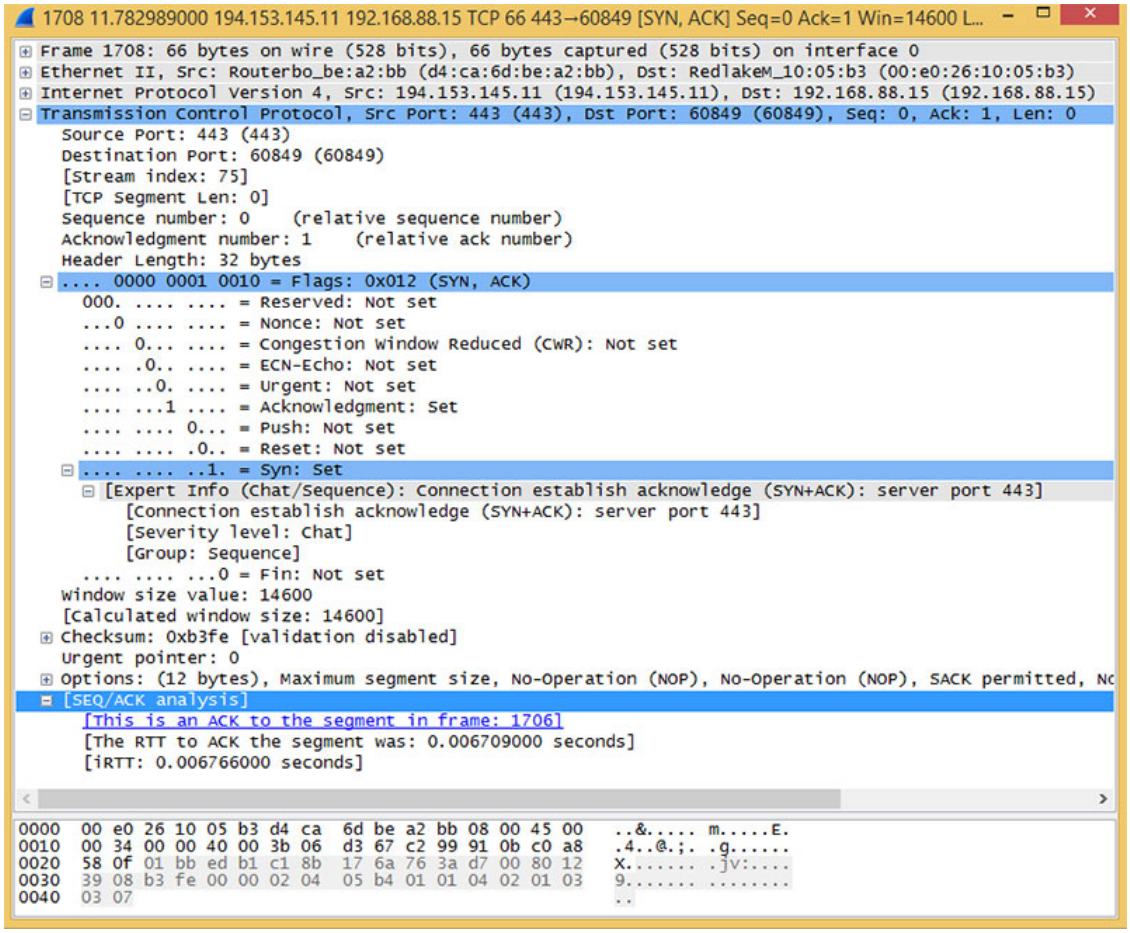


фигура 5 *Псевдохедър*

На следващите фигури са представени реални TCP сегменти, участващи в една сесия.



фигура 6 *SYN сегмент*

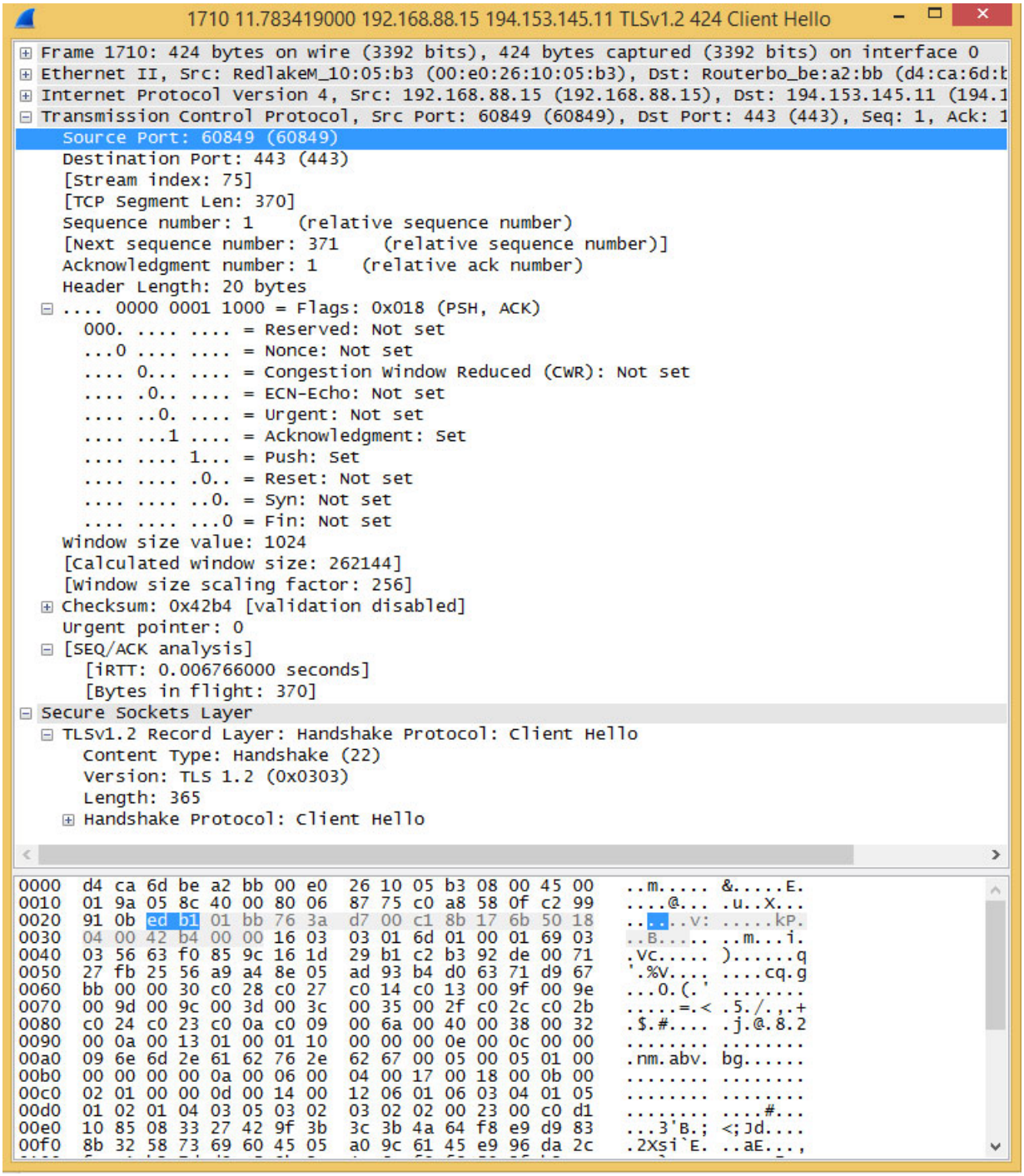


фигура 7 *SYN, ASK сегмент*

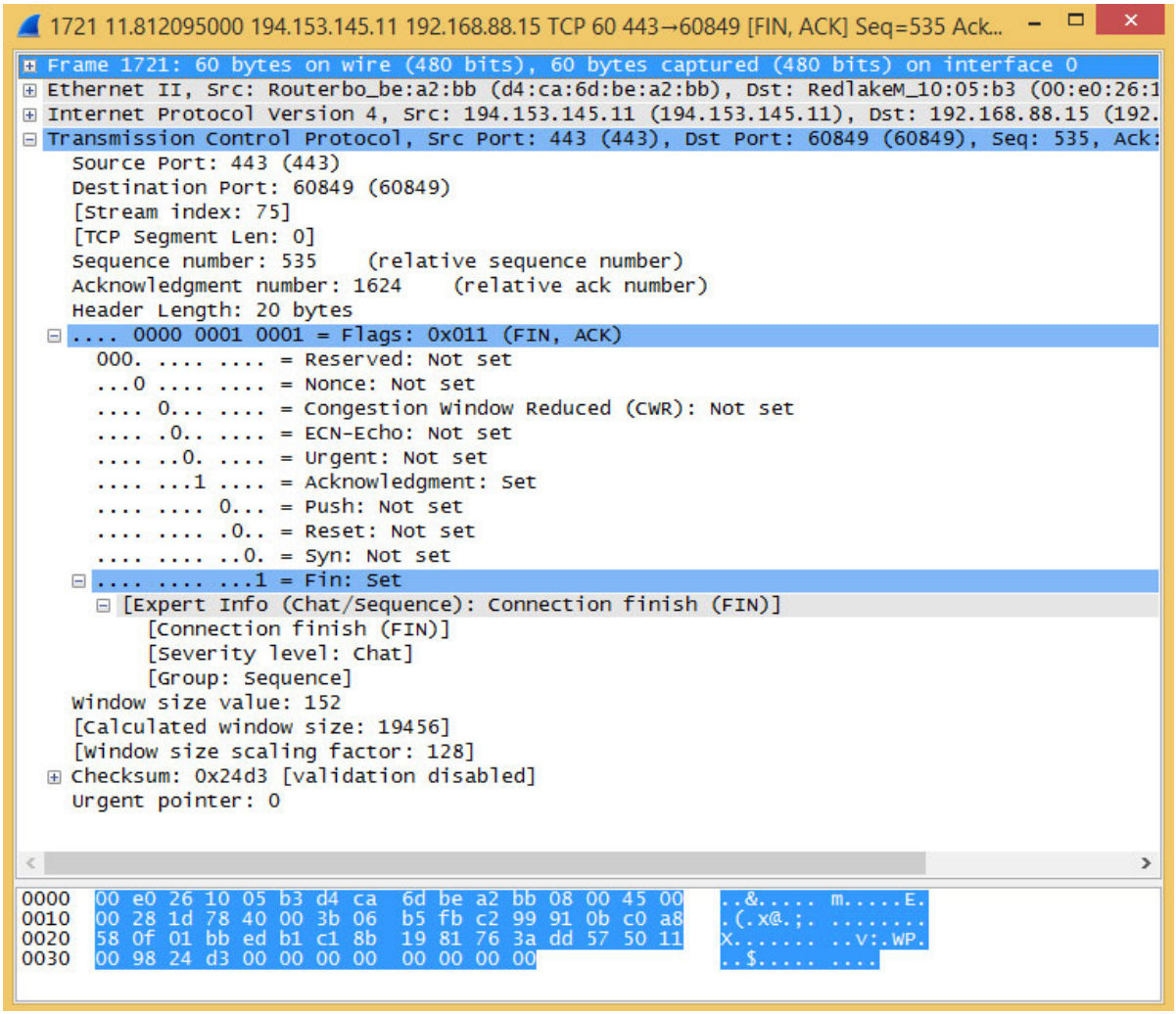
Първият сегмент (фигура 6) е SYN сегмент за инициализиране на TCP сесия. Източникът е хост с адрес 192.168.88.15, а целевият адрес е 194.153.145.11. Вторият сегмент (фигура 7) е [SYN, ASK] сегмент за установяване на сесията. Сегментът от фигура 8 е за трансфер на данни, свързани със SSL протокол.

Стойността на SEQ (Sequence Number) е 1, дължината на сегмента е 370 байта, което означава, че следващият SEQ номер трябва да бъде 370+1=371. Изчисляването на стойностите на SEQ и ACK при комуникацията, отговаря на следното правило: *SEQ=ACK, ACK=1+предадени байтове на TCP сегмента*.

Останалата последователност от сегменти не е представена. На фигура 9 е показан [FIN, ACK] сегмент за приключване на TCP сесията.



фигура 8 *DATA сегмент*



фигура 9 *[FIN, ACK] сегмент*

Класификация на заплахите за сигурността на мрежите

Има няколко метода, които се прилагат за използване на уязвимостите на TCP/IP мрежите. Тези уязвимости могат да се класифицират в следните три категории:

- DoS атаки (Отказ на услуги);
- Разпределени DoS атаки (DDoS);
- Атаки с измами с данни.

DoS атаки

Една често срещана форма на мрежово базирана атака е DoS атаката (Denial of Service - отказ на услуги). При DoS атаките на упълномощени потребители се отказва достъп до мрежови услуги. Обикновено основна мишена на DoS атака са Web сървъри, сървъри за приложения и

комуникационни връзки. Хакерите затрупват сървърите и комуникационните връзки с ненужни данни, разстройват достъпа до данните и разбиват компютри. Често срещан пример за DoS атака е използването на FTP протокол от неупълномощен потребител за качване на голям обем данни. Това причинява ненужно блокиране на дисково пространство и генерира мрежов трафик. Като резултат FTP протоколът става неизползваем.

Осъществяването на DoS атака е лесно, защото тя не изисква специални технически умения или достъп до данните в мрежа. В действителност, DoS атаките са различни от останалите мрежови атаки, защото те удрят по мрежовите услуги, правейки ги в края на краищата недостъпни за упълномощени потребители. Другите атаки обикновено се целят в данните в мрежата и не разстройват мрежовите услуги. Освен това, в сравнение с другите атаки, проследяването и идентифицирането на предприелите DoS атака е трудно, защото атакуващите са разположени на различни позиции в мрежата и обикновено се представят с чужда самоличност.

DoS атаките могат да идват от всяка точка на мрежата.

DoS атаките могат да се появят под различни форми, както следва:

- Разстройване на мрежов трафик;
- Разстройване на връзка между два компютъра в мрежата;
- Отказ на услуги от сървъра на клиент.

Като пример може да се даде DoS атака, при която услуги на компонент на мрежата са временно разстроени, в този случай на сървър за принтер. В една организация има три етажа и има инсталиран принтер на мрежата на всеки етаж. Служителите използват принтера без проблеми от много време. Потребителите, използващи принтера на първия етаж, информират мрежовия администратор, че не могат да печатат документите си, когато използват този принтер. При оглед на сървъра за принтера, администраторът открива огромен брой чакащи заявки за печатане. Тези заявки са заели свободното място на сървъра за принтера и отказват достъп на останалите потребители.

Можем да класифицираме DoS атаките в две категории: *наводняващи атаки* и *софтуерни атаки*. В следващата секция ще се запознаете накратко с тези атаки.

Наводняващи атаки

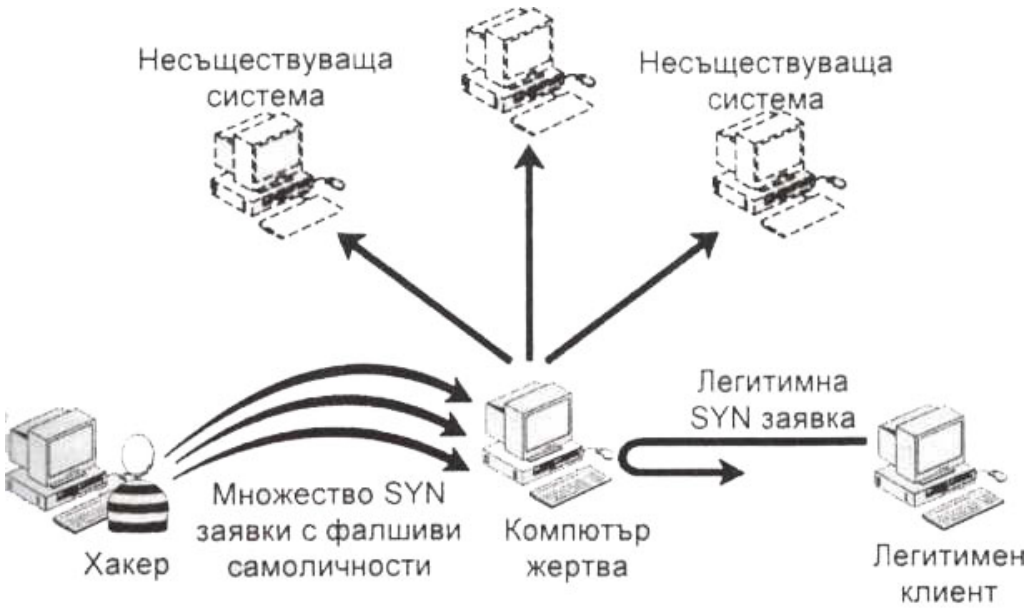
Наводняваща (flood) атака затрупва компютърните ресурси и мрежата с огромен брой фалшиви заявки. Мрежовите устройства, като например рутери и NIC, имат ограничен капацитет за обработване на пакети. Използвайки това, атакуващите изпращат голям брой малки пакети за кратко време с цел претоварване на мрежата. Поради това наводнение, рутерите започват да изпускат легитимни пакети, в опитите си да вървят в крачка и с фалшивите, и с легитимните пакети. Рутерът се натоварва и мрежата се забавя. Трудно е да се засекат наводняващите атаки, защото трафикът на атаките е същият като този на легитимните потребители.

SYN наводняващи атаки

SYN наводняващите атаки използват процеса на трипосочно ръкостискане на TCP (фигура 1). Една такава атака е насочена към използване на TCP връзките на мрежата към сайт, пречейки по този начин на легитимните потребители да се свържат със сайта.

При създаване на връзка със сървъра, клиентът изпраща SYN пакет (съдържащ началния номер на последователност и идентичност на хоста) към сървъра. Сървърът потвърждава получаването с връщане на SYN + ACK пакет към клиента. Връзката е създадена напълно, когато клиентът потвърди с изпращане на ACK пакет на сървъра.

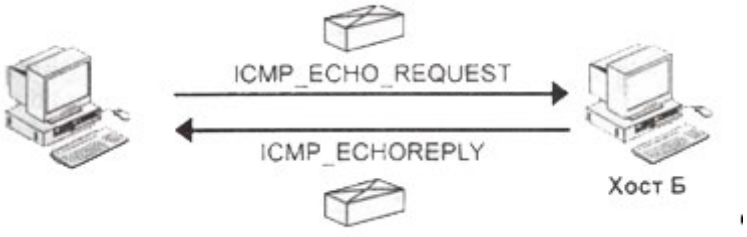
Проблемът започва, ако в SYN пакета се изпрати фалшива идентичност. Заради това потвърждението (SYN + ACK), изпратено от сървъра, никога няма да достигне легитимния клиент. В края на краищата, връзката прекъсва и каналът за заявки на сървъра се освобождава за друга заявка. При SYN атака атакуващият изпраща много фалшиви SYN заявки (с други думи, фалшиви самоличности), което завършва с изпращане на потвърждение (SYN + ACK) от страна на сървъра до несъществуващи клиенти. Следователно сървърът чака потвърждение по 75 секунди на заявка за връзка. Тъй като сървърът вече е зает с чакане, той не отговаря на други заявки от легитимни потребители. Затова сървърът е наводнен със SYN заявки, защото каналът за заявки е свободен (фигура 10).



фигура 10 *SYN наводняваща атака*

SMURF или ICMP_ECHOREPLY наводняващи атаки

SMURF атаките са базирани на *ICMP echo request* и *echo reply* пакети. ICMP означава протокола Internet Control Message Protocol, който е отговорен за поправяне и докладване на грешки, относно доставката на IP пакети. През *ping* услугата ICMP докладва за състоянието на мрежа или хост в мрежата. Ако всичко е наред, отговорът на ICMP се получава от хоста, на който е пусната *ping* услугата. Ако хостът е недостижим, ICMP показва съобщение, дестинацията е недостижима "*destination unreachable*". Когато се стартира *ping* команда, се изпраща ICMP_ECHO_REQUEST до компютъра, чието съществуване трябва да се потвърди. Ако компютърът присъства в мрежата, се връща ICMP_ECHOREPLY от другия край (фигура 11).



фигура 11 *ICMP echo request и echo reply*

При SMURF атака участват три компонента: компютърът на хакера, пакетен усилвател и компютър-мишена (жертва). SMURF атаките започват, когато хакерът подправи адреса на изпращача на *echo request* пакет с адрес

на избран хост-жертва. Тогава той изпраща пакета до разпределен адрес на мрежа - трето лице, която е позната още като мрежа-усилвател. Сега всички хостове от мрежата- усилвател изпращат *echo reply* пакет към компютъра-жертва. Това се отразява не само на компютъра-жертва, но и на мрежата-усилвател.

фигура 12 *SMURF атака*

Крехки или UDP Flood Атаки

Крехките (fraggle) атаки или UDP flood атаките са подобни на SMURF атаки. Също както при SMURF атака се разпращат *ICMP echo request* пакети по мрежата, крехката атака разпраща UDP пакети по мрежата. Тези пакети съдържат фалшиви IP адреси на други компютри от мрежата. Всички хостове, на които е изпратена *UDP echo* заявка, отговарят на другия компютър от мрежата.

Софтуерни атаки

Софтуерните атаки се възползват от присъщи на софтуера уязвимости. При този тип атаки хакерите създават пакети с фалшива информация, за да използват слабости на софтуера.

Примери: Компютър от мрежата може да не е конфигуриран добре. Хакер, който знае за това, може да се опита да преправи или унищожи информацията за конфигурацията на този компютър. Като резултат, други компютри от мрежата може да не успеят да получат достъп до този компютър. По същия начин, поради слабости при производството на рутер хакерът може да има възможност да промени данните в таблицата за рутиране. Резултатът от това може да е спиране на функционирането на мрежата.

Софтуерните атаки могат да се предотвратят с инсталиране на “кръпки” или пачове (patches) за софтуера, които премахват уязвимостите. В

защитната стена могат да се добавят правила за филтриране на деформирани пакети, преди да достигнат компютъра-мишена.

DoS софтуерни атаки

Ping на смъртта или ICMP_ECHO_REQUEST атака

Атаката “ping на смъртта” е базирана на ping услугата на TCP/IP. При тази атака хакерът изпраща верига от фрагментирани ping команди, съдържащи големи по размер ICMP пакети, до компютъра-мишена. Максималният размер на пакет, който може да се изпрати през ping услугата, е 64 KB, така че хакерът изпраща известен брой ICMP пакети (от по най-малко 64 KB) чрез хиляди и милиони ping команди за секунда. Повечето компютърни системи не могат да обработят и възприемат пакет, който надвишава максимално приетият размер за IPv4 протокола – 65 535 байта. Получаването на по-голям от този размер пакет обикновено причинява срив в системата-жертва. Хакерът може също така да промени размера на пакетите на повече от 64 KB. С цел да отговори на ping заявките, когато хостът-мишена започне да сглобява пакетите, става ясно, че те са прекадено големи за буфера му. Следователно се получава препълване на буфера, защото компютърът-мишена е прекадено зает да сглобява толкова много пакети и да изпраща обратно отговори със същите размери.

Целта на DoS-базираната команда “ping на смъртта” е да наводни компютъра-жертва с много на брой заявки и така да препълни трафика на отдалечения сайт. Това се превръща в отказ на услуги от компютъра-жертва към другите компютри по мрежата. За да изпълнят успешна команда “ping на смъртта”, хакерите трябва да имат значително по-голям трафик (да поддържат многобройни ping команди) в сравнение с трафика на компютъра-жертва.

Следва пример за ping команда, която изпраща свръхголеми пакети на хоста- жертва:

ping -1 6500 -s 1 {IP адрес на компютъра-жертва}

В предходната команда, -1 6500 задава размера на пакета на 6500, а -s определя времето за броене на хоповете (hops).

Можете да предотвратите изпълнението на командата “ping на смъртта”, като забраните фрагментирани ping команди на вашата система. Това позволява преминаването само на обикновени 64-байтови ping пакети

през системата ви и блокира фрагментираните ping пакети, които са с по-голям размер.

Атаки на услугата DNS

- **DNS измами**. Тези атаки са познати и като DNS превземане. Хакерът има достъп до DNS услуги и променя информацията, която свързва домейн име и IP адрес. Заради това потребителите биват пренасочвани към различен сайт от този, който са искали.
- **DNS препълване на хост име**. В отговора на DNS за хост име има дефинирана максимална дължина на хост името. DNS препълването на хост име се получава, когато се провали проверката на разрешената дължина на името в DNS отговора. Заради това се получава препълване на буфера, когато хост името е копирано в DNS сървъра.
- **DNS препълване на дължина**. Освен максимална дължина на хост име, DNS отговорът още съдържа и поле за дължина, което има допустим предел от 4 байта. Ако бъде определена голяма стойност, се получава препълване на буфера, което позволява на атакуващия да изпълни администраторски команди на системата-мишена.
- **DNS трансфер на зони от привилегировани портове (1-1024)**. Този тип препълване се получава, когато хакерът използва DNS клиентско приложение, за да пренесе зони от вътрешен DNS сървър през привилегировани портове (1-1024). DNS зоната дефинира DNS йерархично пространство за имена, които се използват за разграничаване на различните области на действие на DNS сървъра. С други думи, те дефинират кои DNS сървъри имат право да разрешават въпроси за анализ на имена за дадена секция от DNS йерархията.

DDoS атаки

DoS и DDoS атаките имат подобни имена. Но значи ли това, че те действат по подобен начин? Каква е разликата между DoS и DDoS атака?

В обикновена мрежово базирана DoS атака хакерът използва инструменти-приложения, за да наводни компютъра-мишена с многобройни пакети от данни. Резултатът от това е, че упълномощени

потребители не получават достъп до услуги на компютъра-жертва или може би до цялата мрежа. При такива атаки хакерите използват измамни или несъществуващи IP адреси, за да скрият идентичността си.

Както и DoS атаките, DDoS атаките също целят отказа на мрежови услуги на легитимни потребители. Разликата обаче е в броя на компютрите, замесени в атаката. В DoS атака има замесени само 2 компютъра, тези на хакера и жертвата. В случая на DDoS атаката обаче може да има само един хакер, но ефектът на атаката се умножава от използването на няколко атакуващи агента по мрежата (в действителност по Интернет).

Начин на действие на DDoS атаките

При DDoS атака, процесът на атакуване започва, когато хакерът пробие в няколко компютъра през мрежа. Той прави това, използвайки характерни уязвимости компютрите или мрежата.

Нека компютрите, използвани в DDoS атаките, наричаме компрометирани, а компютърът, който в края на краищата ще бъде атакуван, ще наричаме компютър-жертва.

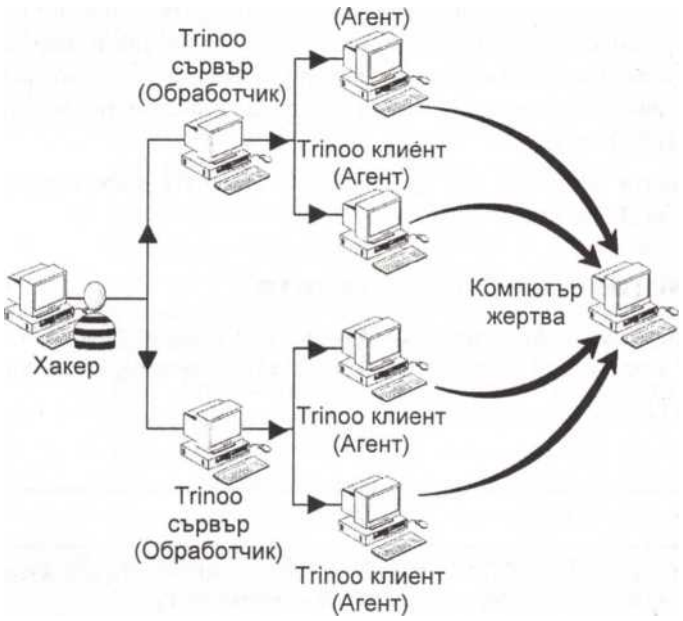
След като придобие достъп до компютрите, хакерът първо инсталира софтуер (например инструменти за заместване), който прикрива следите за съществуването му по компютрите.

Хакерът инсталира малко по-специализирани приложения, които му позволяват отдалечено да контролира компрометираните компютри и да задава команди по мрежата, които на свой ред да насочат атаката срещу компютъра или сайта-жертва. Целият процес на атаката, обяснен тук, е силно автоматизиран.

Бдителният хакер обикновено започва с пробиване в няколко компютъра. През тези компютри той пробива в още няколко, повтаряйки процеса няколко пъти. Този повтарящ се процес подготвя база за DDoS мрежа, която съдържа хиляди компютри.

За да насочи атаката, хакерът пуска една-единствена команда от собствения си компютър, изпращаща команден пакет до всички компрометирани компютри, която ги инструктира да насочат определена атака срещу определена жертва. Когато хакерът реши да спре атаката, той изпраща друга единична команда.

Много източници категоризират компрометираните компютри като обработчици и агенти. Обработчиците са компютрите, до които хакерът изпраща командата най-напред. Тези компютри после пренасочват командата към компютър-агент, който в края на краищата изпълнява атаката.



фигура 13 *SMURF атака*

Инструменти за DDoS

Има няколко достъпни инструмента за предприемане на DDoS атаки. Повечето от тях използват разпределена технология, за да създадат голяма мрежа от хостове.

Пример за инструмента за DDoS: Trinoo и Tri Flood Network (TFN).

Trinoo

Trinoo се използва при UDP Dos наводняващи атаки от различни канали по мрежата. Този инструмент по същество създава мрежа, която се състои от няколко сървъра и голям брой клиенти. Сървърите в мрежата са в категорията Trinoo сървъри, а клиентите – Trinoo клиенти.

При DoS атака, която действа в Trinoo мрежа, компютърът на хакера е свързан с главния компютър в мрежата (Trinoo сървъра). Чрез изпращане на една единствена команда компютърът на хакера указва на Trinoo сървъра да започне DoS атака срещу един или повече IP адреса. Trinoo сървърът, от своя

страна, предава контрола на другите Trinoo агенти, които рано или късно извършват атаката.

Tribe Flood Network (TFN)

Инструментът Tribe Flood Network (TFN) действа подобно на Trinoo. Като при Trinoo, TFN също се използва за осъществяване на DDoS атаки срещу няколко мишени по мрежата. Единствената разлика е, че TFN може да използва фалшиви IP адреси на източника. TFN атаката може да се използва за осъществяване на следните DoS атаки:

- UDP наводняващи атаки;
- TCP SYN наводняващи атаки;
- Атаки “ping на смъртта”;
- SMURF атаки.

Като Trinoo атака, при TFN атаката компютърът на хакера е свързан с главния компютър в TFN мрежата (TFN сървър). Чрез изпращане на единична команда хакерският компютър указва на TFN сървъра да започне DoS атака срещу един или повече IP адреси. TFN сървърът, от своя страна, предава друга команда до TFN клиентите, които накрая осъществяват атаката. Освен това комуникацията между TFN обработчиците и агентите е криптирана.

Атаки с измами на данни

Атаките с измами на данни включват прихващане, промяна и повреждане на данни, пътуващи по мрежата или намиращи се на хостовете. Тези атаки може още да включват препращане на пакети от данни по мрежата или променяне на маршрута на пакетите от данни до невалидни получатели. Хакерите изпълняват тази атака чрез използване на характерни уязвимости в мрежовите протоколи или в операционните системи. Често срещана атака с измама на данни е тази, при която хакерът получава достъп до Web сайт и променя съдържанието му.

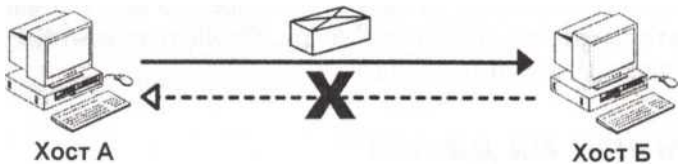
Има различни методи, като например *измами* (spoofing), *подслушване* или слухтене (sniffing), превземане на сесии, промяна на маршрута и сканиране на портове, чрез които може да се създаде атака с измама на данни. В действителност, можете да категоризирате много от тези методи

като ефикасни инструменти за наблюдаване на мрежов трафик и улавяне на пакети от данни по мрежата.

Измама

Измамата (spoofing) е атака, при която компютърът на хакера се маскира като компютър от мрежата на компютъра-мишена. Целта на компютъра на хакера е да измами компютъра-мишена, че е оригиналният компютър, с който мишената се предполага, че взаимодейства. Намерението е компютърът-мишена да бъде подмамен да изпрати или сподели данни или да получи права за промяна на данните.

Измамата може да бъде сляпа или активна. При сляпата измама хакерът не може да види отговорите, изпратени от компютъра-мишена. Причината е, че хакерът няма пълната информация за условията в мрежата, което означава, че вероятно няма IP адреса на компютъра, за който иска да се представи, или пък няма права за достъп, които компютрите по мрежата споделят. В такава ситуация хакерът използва всички възможни техники, за да получи достъп до мрежата. Това прилича на хвърляне на стрелички в тъмното.



фигура 14 *Сляпа измама*

При активната измама хакерът има информация за правата за достъп, споделяни между компютъра-хост (за който иска да се представи) и компютъра-мишена. Тази информация помага на хакера да види отговорите от компютъра-мишена. Заради това данните могат лесно да бъдат повредени, променени и изпратени на други получатели по мрежата.

фигура 15 *Активна измама*

Атаките с измами по мрежата могат да се класифицират в следните категории:

- IP измама;
- ARP измама.

IP измами

IP измамата е метод, при който хакерът получава достъп до компютъра-мишена чрез използване на измамен IP адрес на доверен хост. Хакерите изпълняват IP измама чрез използване или на сляпа, или на активна измама.

Атаката с IP измама (наречена още измама с отгатване на последователността на IP пакетите) се прави по време на процеса на свързване чрез трипосочно ръкостискане. За да започне атака с IP измама, хакерът първо трябва да фалшифицира IP адреса на доверен хост в мрежата. След това трябва да заяви пореден номер на компютъра-мишена, като по това време той записва началния пореден номер в заглавната част на информационните пакети с данни. Тази задача е много сложна, защото когато мишената изпраща начален пореден номер като потвърждение, хакерът трябва да отговори правилно, като познае успешно TCP началния пореден номер.

Протокол ARP и измами с него

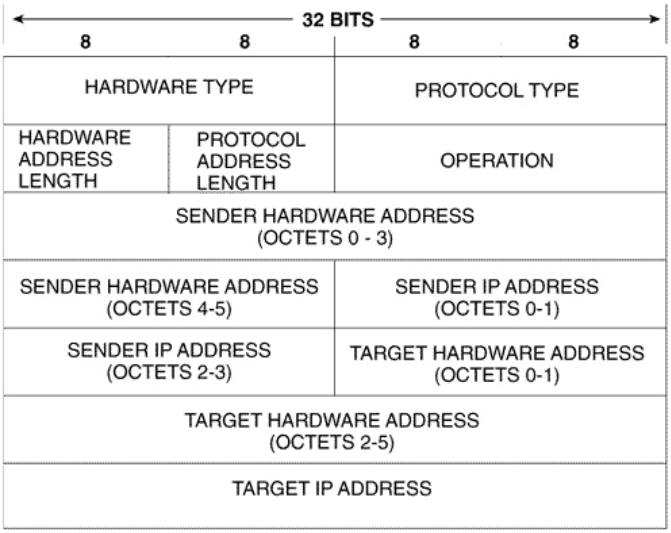
ARP измамата е атака, при която хакерът променя таблицата на протокола ARP (Address Resolution Protocol) чрез експлоатиране на комуникацията между IP и Ethernet протоколите. Атаката с ARP измама може да се осъществи по Ethernet мрежи, използващи TCP/IP протоколи.

Функциониране на ARP (Address Resolution Protocol)

ARP е протокол за преобразуване на адреси. Превръща логическите адреси (например 32-битови IP адреси) от мрежовия слой във физически адреси от канален слой (например MAC адреси) [RFC 826]. За целта всеки хост поддържа ARP кеш таблица, където съхранява съответствията между IP и MAC адрес, научени динамично по време на комуникацията с други хостове или въведени статично от администратора на системата. ARP използва броадкасти до хостовете в локалния сегмент за определяне на дадено съответствие, което добавя като запис в ARP таблицата на хоста за бъдещо използване. Валидността на записите може да се контролира чрез няколко механизма:

- таймаут – при добавяне на запис в таблицата се определя време на валидност, след което той се премахва;
- периодични уникаст запитвания – изпращат се периодични уникаст запитвания към регистрираните хостове. Ако отдалеченият хост не отговори, записът се премахва от кеша;
- уведомяване от протокол – ако протокол от по-горен слой установи проблеми при доставката, той уведомява активния ARP процес в хоста, който от своя страна премахва записа за отдалечения хост от таблицата му.

Структурата на заглавната част на ARP и описанието ѝ е показано на фигура 16.



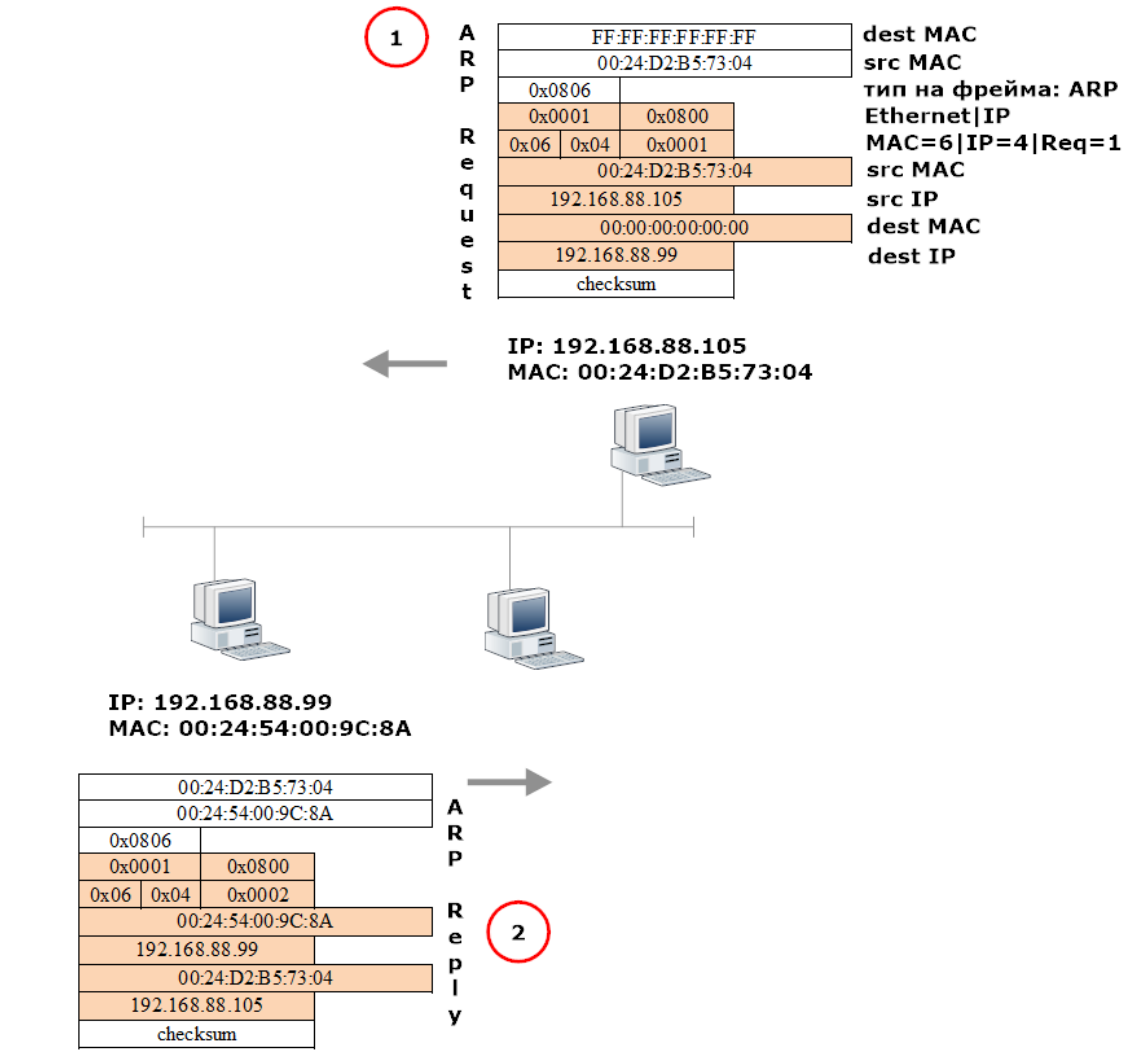
фигура 16 *Заглавна част на ARP*

Hardware type – 2-байтово поле, идентифициращо типа на хардуера (например Ethernet, Token-Ring или друг тип мрежа). За Ethernet това поле има стойност 0x0001;
Protocol type – 2-байтово поле, идентифициращо типа на протокола от мрежовия слой (например IP=0x0800, IPX=0x8137);
Hardware length – 1-байтово поле, задаващо дължината в байтове на хардуерния адрес (например Ethernet има 6-байтов MAC адрес);
Protocol length - 1-байтово поле, задаващо дължината в байтове на протоколния адрес от мрежово ниво, който се преобразува. Например за IP=0x0800 полето трябва да съдържа стойност 4;
Operation - 2-байтово поле, идентифициращо типа на извършваната операция:
0x0001=ARP request;
0x0002=ARP reply;
0x0003=RARP request;
0x0004=RARP reply.
За разграничаване на типа на кадъра (ARP или RARP), в полето Ethertype на DLC хедъра (фигура 17) се записва 0x0806 (ARP) или 0x8035 (RARP).

Sender hardware address – 6-байтово поле за хардуерния адрес на изпращача;
Sender protocol address – 4-байтово поле за логическия адрес на изпращача;
Target hardware address - 6-байтово поле за хардуерния адрес на получателя;
Target protocol address - 4-байтово поле за логическия адрес на получателя.

	Ethernet II хедър					
байта	6	6	2	28	18	4
	DA	SA	**0x0806**	ARP Request, ARP Reply	Padding	FCS

фигура 17 *DLC ARP кадър*

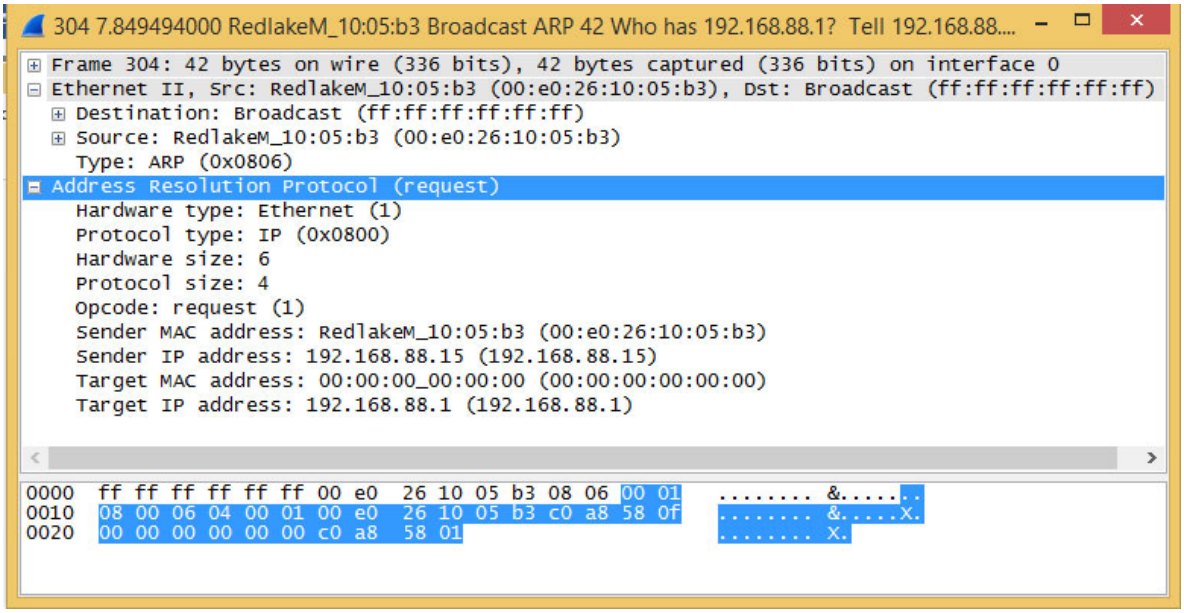


фигура 18 *ARP заявка и отговор*

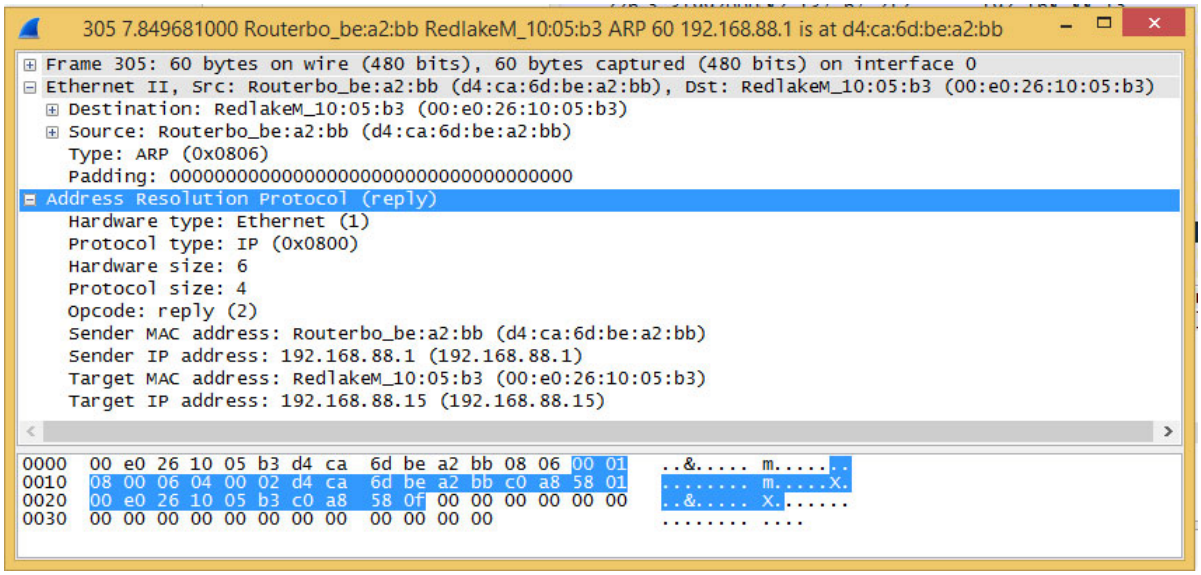
На фигура 18 е представена ARP заявка (Operation=0x0001) и ARP отговор (Operation=0x0002). Компютър с IP адрес 192.168.88.105 изпраща в локалния сегмент ARP броадкаст (destMAC=FF:FF:FF:FF:FF:FF) за

установяване на непознат MAC адрес, съответстващ на локален IP адрес 192.168.88.99. Търсеният хост отговаря на запитването чрез ARP отговор като предоставя физическия си адрес на хоста-изпращач. Втората възможност е, ако изпращащият хост е установил, че получателят не се намира в същия сегмент. Тогава неговият ARP броадкаст ще бъде предназначен за установяване на физическия адрес на локалния шлюз (ако не го знае все още), който да препрати пакета до хоста-местоназначение.

Подобен пример с два последователни фрейма за откриване на MAC адреса на рутер е показан на фигура 19 и фигура 20.

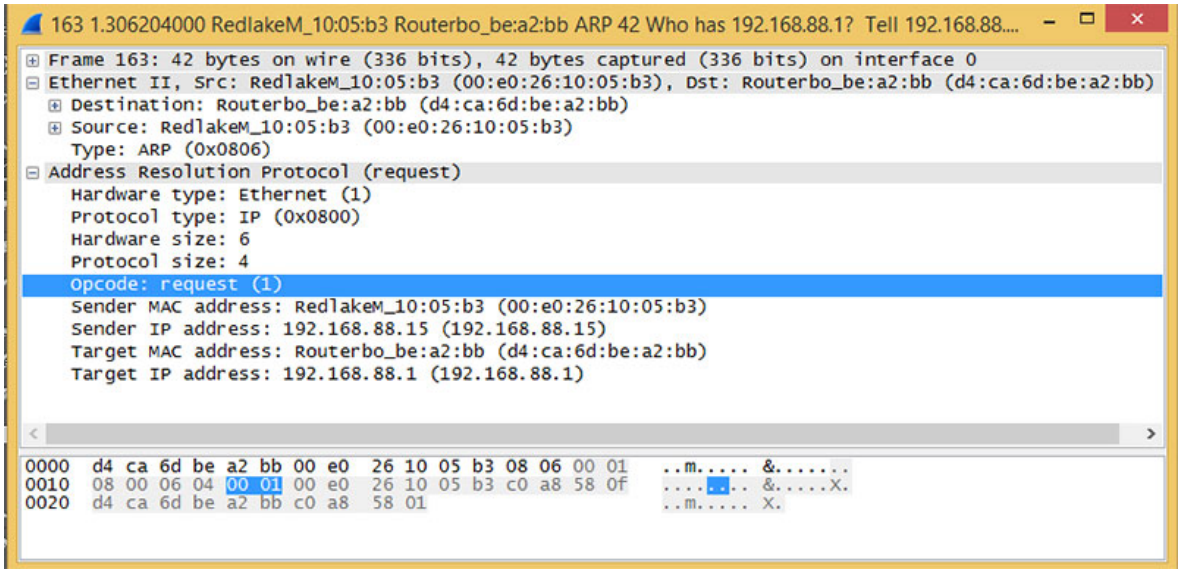


фигура 19 *ARP заявка*

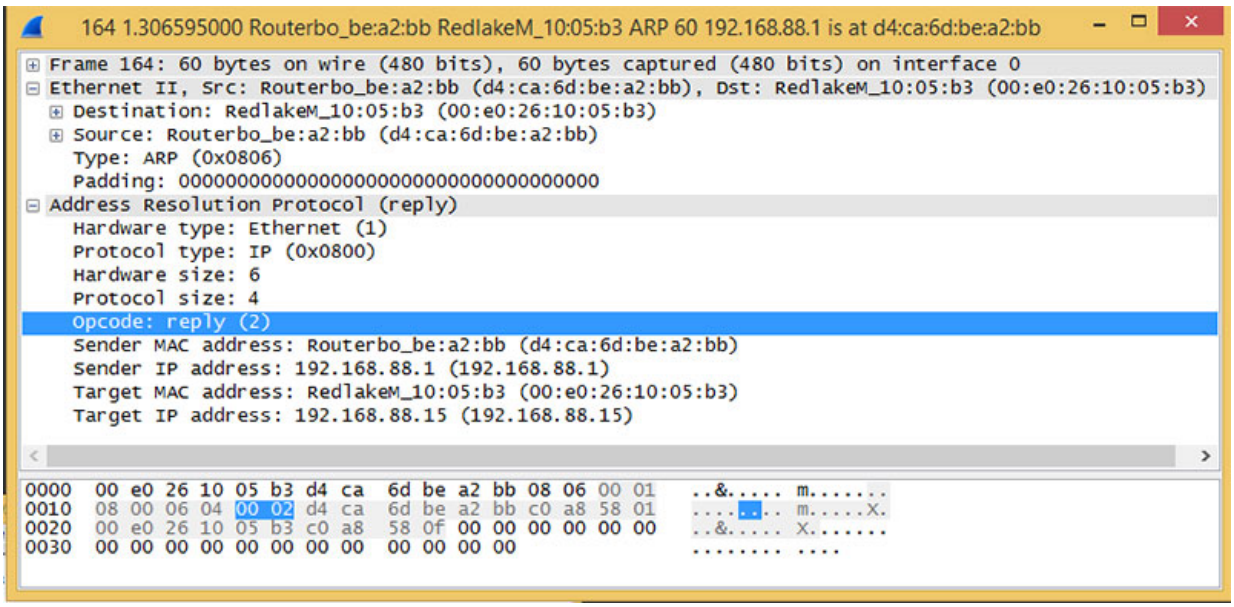


фигура 20 *ARP отговор*

Пример за периодично уникаст запитване е представен на фигура 21 и фигура 22. Типът на полето Opcode, на фигура 21, е със стойност 0x0001 (ARP request), а за фигура 22, е 0x0002 (ARP reply). За разграничаване типа на кадъра (ARP или RARP), в секцията Ethertype II, полето Type заема стойност 0x0806 (ARP). Преобразуването се извършва между адреси от хардуерен тип Ethernet (Hardware type: 0x0001) с дължина 6 байта (Hardware size: 6) и протокол IP (Protocol type: 0x0800) от мрежовия слой с дължина на адреса 4 байта (Protocol size: 4).



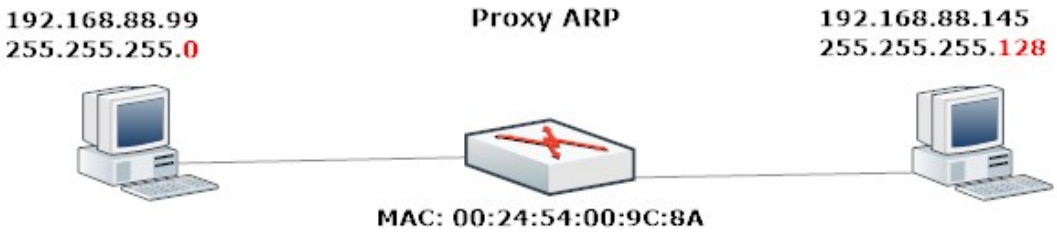
фигура 21 *ARP заявка*



фигура 22 *ARP отговор*

В някои случаи е възможно използването на Proxy ARP, което позволява на друго устройство (например маршрутизатор) да отговаря на

ARP запитвания от името на отдалечен хост, намиращ се в друга подмрежа. Това осигурява възможност за прозрачна комуникация между отдалечени хостове. Например, ако даден хост-изпращач е конфигуриран неправилно и се объркa при изпращането на ARP запитване към хост извън локалния сегмент, то маршрутизаторът, конфигуриран като Proxy ARP, ще отговори със собствения си физически адрес. Това позволява пакетът да бъде препратен след това към правилния хост-местоназначение.



фигура 23 *Proxy ARP*

Подобен пример е представен на фигура 23, където хост 192.168.88.99 от мрежа 192.168.88.0 иска да комуникира с хост 192.168.88.145 от мрежа 192.168.88.128. Мрежовата маска на първия хост (255.255.255.0) е сгрешена. Това го обърква при пресмятането на принадлежността на адрес 192.168.88.145. За него този адрес е част от локалния сегмент и затова изпраща локален ARP броадкаст за установяване на физическия му адрес. Това запитване не стига да хоста-местоназначение и той не отговаря. Маршрутизаторът, конфигуриран като Proxy ARP, приема това запитване и отговаря от името на целевия хост като изпраща собствения си хардуерен адрес. Изпратеният към него пакет бива препратен към хоста-местоназначение.

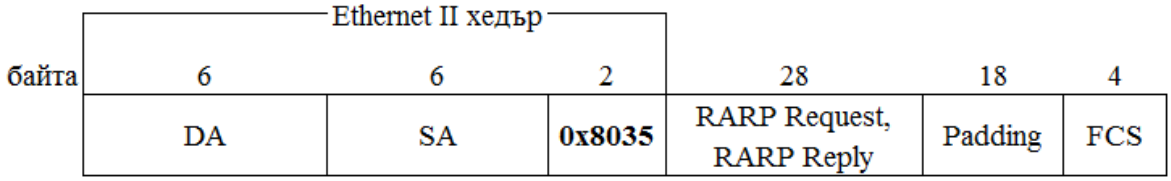
RARP (Reverse Address Resolution Protocol)

RARP е протокол за динамично преобразуване на физическите адреси на хостове в логически адреси от мрежово ниво (например IP адреси). Той извършва противоположно действие на ARP протокола. Използва същия формат на хедъра, както ARP протокола. Кадърът, пренасящ RARP, се разпознава по типа на полето Ethertype=0x8035 (фигура 24).

Протоколът е предназначен за назначаване на логически адреси на бездискови станции. За целта се използват RARP сървъри, където в статични таблици се съхраняват съответствията между хардуерни и логически адреси. За поддържането им се грижат мрежови администратори.

Подобно на ARP и този протокол използва броадкасти. Това означава, че във всеки локален сегмент трябва да има поне по един RARP сървър, който да отговаря на такъв тип запитвания. Недостатъците на RARP са свързани с необходимостта от:

- такъв тип сървъри за всеки един сегмент, поради невъзможността маршрутизаторите да препращат RARP броадкасти;
- администратор за създаване и поддържане на статичните асоцииращи таблици.



фигура 24 *Ethertype=0x8035*

RARP не е единственият протокол за преобразуване на хардуерни в логически адреси. Представители на този тип асоцииране са още BOOTP и DHCP, предназначени да заменят RARP. Всеки един от тези протоколи се стреми да отстрани проблемите, съществуващи при предшественика му. В момента актуалният протокол е DHCP.

ARP измами

Атаката с ARP измама включва превземане на разпращани съобщения, промяна на IP адреса на получателя с IP адреса на хакера и отговор с MAC адреса на хакерския компютър. След като подателят получи MAC адреса, той приема, че полученият MAC адрес е верен и предава данните на компютъра на хакера. Този процес на измама е още познат като ARP отравяне.

Подслушване

Подслушването (sniffing) е атака, целяща улавяне на мрежова информация. Атаката включва поставянето на подслушващи устройства на различни входни точки по мрежата. Тези входни точки могат да са компютри, рутери, мрежови сегменти, свързани с Интернет, или мрежови сегменти, свързани със сървъри, които получават пароли.

Подслушвателят е програма, която надзирава, ръководи и следи преноса на данни по мрежата. Основната цел на подслушвателя е да

подслушва мрежов трафик. Подслушващите устройствата имат следните компоненти:

- *Драйвер за улавяне (capture driver)*. Драйверът за улавяне прихваща мрежов трафик от кабелен Ethernet, филтрира мрежовия трафик за информацията, която искате, и съхранява филтрираната информация в буфер.
- *Буфер*. Буферът е софтуерна програма, която помага за съхраняване на данните, уловени по мрежата. Данните могат да се съхраняват по два начина: докато буферът се запълни с информация или по метода "всеки срещу всеки" (round-robin), при който данните се изтриват, когато достигнат до определено ниво в буфера. Това означава, че данните в буфера винаги се заменят с новоуловени такива.
- *Декодер*. Данните, предавани по мрежата, не пътуват в нормалния си четим вид. Те пътуват в двоичен формат, който трябва да се декодира. Декодерите се използват за декодиране на двоичните данни в мрежата, за да могат те да се интерпретират и покажат в четим вид.

Ethernet мрежата е построена на споделени принципи. Когато данните се предават по Ethernet мрежата, заявките достигат до всички компютри в мрежата (освен ако не е разделена на сегменти). Въпреки че всички компютри получават заявката, отговаря само компютърът, за който е предназначен пакетът от данни. Компютрите, които не са действителни получатели на пакетите от данни, игнорират заявката.

Подслушвателят спира филтрирането и пуска Ethernet хардуера (NIC) да работи в смесен режим, да приема всички пакети от данни, независимо дали пакетите са били предназначени за него или не.

Промяна на маршрута (rerouting)

Рутерът е друга уязвима входна точка в мрежата, която позволява неупълномощен достъп до пакети от данни. Рутерът има таблица, съдържаща информация за съставянето на маршрути и конфигурации на другите хостове от мрежата. Протоколът RIP (Routing Information Protocol) се използва за поддръжка на таблицата за маршрутите. Хакерите използват протокола RIP, за да манипулират информацията за съставяне на маршрутите или конфигурациите на хостовете в таблицата с маршрутите.

Манипулациите на таблицата с маршрутите могат да завършат с неправилни конфигурации на маршрути на хостове по мрежата, което може на свой ред доведе до загуба на пакети от данни по мрежата.

Сканиране на портове

Сканирането на портове е популярна техника за разследване, използвана от хакерите за проникване в мрежа или хост от мрежа. Всички компютри, свързани в мрежата или към Интернет, слушат всички комуникации по едни или други портове. Тези портове действат като врата, през която данните влизат или излизат. Например TCP и UDP портовете се използват от приложенията за създаване на сесии между възли на TCP/IP мрежи.

При сканиране на портове хакерът опитва да открие всички възможни отворени портове на мрежа и след това изпраща съобщения до всеки един от тях. Има различни инструменти за сканиране на портове. Тези инструменти проверяват всички варианти на TCP и UDP портове на отдалечен хост. Използваната техника може да е различна в зависимост от топологията на мрежата, системата за откриване на пробиви IDS и особеностите на регистрирането в отдалечената машина.

Примерни техники при сканирането на портове, използвани от хакери:

- Vanilla - при тази техника хакерът се опитва да се свърже с възможни 65535 порта;
- Strobe - по-конкретизирана техника за сканиране, при която хакерът търси да експлоатира само известни услуги;
- Фрагментирани пакети - при тази техника хакерът действа чрез изпращане на малки пакети;
- UDP - при тази техника хакерът търси отворени UDP портове;
- Sweep - при тази техника скенерът се свързва с един и същ порт от няколко компютъра;
- Неявно сканиране (stealth scan) - скенерът блокира сканирания компютър, за да не запише действията по сканирането на портове.

Лекция 4

Преглед на защитните стени (firewalls)

Нека да започнем обяснението за защитните стени с един основен пример. Офисът на една организация се посещава от няколко човека. Но само на служители и клиенти на организацията е разрешено да влизат в сградата на офиса. Може би охраната проверява документите за самоличност на всеки, който посещава офиса, преди да му позволи да влезе. В някои случаи също се прави проверка на служителите, за да се провери дали те не внасят предмети, които не им е позволено да внасят (като например домашни любимци и алкохол). Но тези ограничения не предотвратяват смесването на служителите с други хора, външни за офиса решено е на служителите да излизат, но не на всеки е разрешено да влиза в офиса. Така че охраната или всяка друга единица, отговорна за ограничаване и контрол на достъпа до сградата на офиса, може да се сравни със защитна стена.

Какво е защитна стена в контекста на мрежите? Мрежовата защитна стена е много добре дефинирана в речника NSA Glossary of Terms Used in Security and Intrusion Detection като “система или комбинация от системи, която подсилва границата между две или повече мрежи”. Казано просто, защитната стена реализира правила за сигурност, които отделят две мрежи от нежелана комуникация.

Мрежовите защитни стени пазят критична информация от опасности и в същото време позволяват преминаването на информация. Защитните стени предпазват мрежите от хакери и атаки, като например DoS атаки или атаки “ping на смъртта”, и също позволяват преминаването на HTTP пакети и e-mail съобщения.

На кой слой работи една защитна стена?

Една защитна стена може да работи на различни нива в разделена на слоеве мрежова архитектура на OSI и TCP/IP моделите. Най-ниското ниво, на което може да работи една защитна стена, е мрежовият слой. На мрежовия слой защитната стена може да определи дали един пакет идва от автентичен източник. Но на този слой тя не може да провери автентичността на информацията, която се предава.

Защитните стени, които работят на следващия слой, а именно транспортния слой, предоставят по-сложни функции от защитните стени от мрежовия слой. Защитните стени от транспортния слой могат да потвърждават връзки, преди да решат дали да разрешат или не някой пакет. Тази валидност се проверява чрез преглед на определен набор от правила, конфигурирани в самата защитна стена.

Защитните стени, работещи в приложния слой, могат да изпълняват по-сложни функции, като например филтриране на трафика, на базата на съдържанието на пакетите от данни или по-скоро на типовете приложения. Тези защитни стени също документират дейностите на пакетите от данни, които отказват. Защитните стени могат да реконструират пакетите, които са отказали, на базата на записите, които се поддържат.

Функционалност на защитните стени

Някои функции на мрежовата защитна стена, които подсигуряват сигурността и целостта на данните са:

- *Филтриране на пакети*. Става въпрос за филтрирането на входящи и изходящи пакети на базата на информацията за протокола и адреса, въпреки че не се проверява съдържанието на пакета. Това е най-основната функция на една защитна стена.
- *Преобразуване на мрежов адрес*. Тази функция скрива хостове на вътрешни мрежи от хостове на външни мрежи чрез преобразуване адресите и портовете на вътрешни хостове до често срещани външни адреси на защитната стена. Защитни стени, които действат по този начин, предотвратяват наблюдението на вътрешни хостове от злонамерени хостове на публична мрежа.
- *Прокси услуги*. Те предоставят обмен на данни от страна на клиентските приложения с отдалечени системи. Така клиентският компютър се скрива зад защитната стена и за отдалечената система изглежда сякаш проксито (proxy) взаимодейства с нея.
- *Потребителска автентикация*. Тази функция е удобна, когато отдалечени потребители (в публична мрежа) използват динамични IP адреси, за да се свържат с частни мрежи. В този случай ограничение на базата на IP адреси не е практично, защото

потребителят ще получи различен IP адрес, когато се свърже. Затова защитната стена ще поиска проверка на автентичността, преди да позволи влизане в частната мрежа.

- *Тунелиране (Tunneling)*. Чрез създаване на виртуален тунел се позволява на физически отделни мрежи да използват Интернет като среда за комуникация. Една такава реализация на защитните стени помага за създаване на виртуални частни мрежи (Virtual Pi Networking, VPN).

На базата на функционалността, която може да предлагат, защитните стени могат да се класифицират на:

- защитни стени, филтриращи пакети;
- защитни стени, преобразуващи мрежови адреси (Network Address Translation, NAT);
- circuit relay защитни стени;
- прокси защитни стени за приложения.

Защитни стени, филтриращи пакети

Защитните стени, филтриращи пакети (packet-filtering firewalls), са най-основните защитни стени, които работят в мрежовия слой на OSI и TCP/IP модела. Те филтрират пакети на базата на правила, дефинирани в самите тях. Ако пакетите не отговорят на определените критерии, те просто биват отказани: Филтрирането на пакети може да се осъществи в рутерите или през мрежова операционна система (Network Operating System, NOS) с възможности за рутиране.

Защитните стени, филтриращи пакети, имат някои характерни проблеми, защото не предоставят пълна сигурност за вътрешна мрежа. За да се противопоставят на този недостатък, филтрите на пакети обикновено са комбинирани с прокси сървъри и преобразуватели на мрежови адреси.

Стандартните филтри на пакети определят неприкосновеността на пакетите на базата на информацията, съдържаща се в заглавните части на индивидуалните пакети. Теоретично филтрите могат да бъдат конфигурирани да сортират пакети на базата на всяка част от информацията в полетата за данни в заглавието на протокола. Най-честите полета са:

- *Поле за IP протокол (IP protocol field)*. Има четири протокола, UDP, TCP, ICMP и IGMP, срещу които полето за IP протокол може да се използва за филтриране на пакети данни. Но тези протоколи са основни и повечето сървъри и рутери трябва да ги оставят отворени за взаимодействие. Затова филтрирането на базата на полето за IP протокол като цяло не е успешно. Пример за филтриране на IP протокол е сървър, чиято основна цел е да поддържа TCP-базирана услуга, като например HTTP, блокираща всички UDP и ICMP услуги.
- *Филтриране по IP адрес (IP address filtering)*. Това ограничава връзките до (или от) определени хостове и мрежи на базата на IP адресите им. Повечето защитни стени работят чрез определяне на правила като например "*разреши пакети от 172.17.10.11 до 172.17.10.30, но блокирай всички останали пакети*", "*пакети, идващи от 128.162.11.14, нямат разрешение да преминат*" или "*разреши всички пакети без пакети с IP адрес от 172.17.10.11 до 172.17.10.30*".
- *TCP/UDP порт*. Това е най-често използваната информация за филтриране на пакети от данни. Този тип филтриране също се базира на информация, съдържана в полетата данни в заглавието на IP протокола, защото информацията за TCP или UDP номерата на портове е определена там. Често срещани протоколи, които могат да се филтрират на базата на TCP или UDP полето за порт, са Echo, Quote, FTP, telnet, SMTP, DNS, HTTP, Gopher, NetBIOS сесия, SNMP, POP, NFS и X Windows. Някои от тези протоколи са чувствителни към атаки поради високото ниво на оперативен контрол, което дават на атакуващите. Затова е важно да се блокират портовете на тези протоколи.
- *Telnet*. Ако портът на telnet е оставен отворен на хост, това дава на атакуващите отворена врата до командния промпт на машината, което прилича на даване на достъп до цялата машина.
- *NetBIOS сесия*. Ако портът за NetBIOS сесия е оставен отворен за Интернет на Windows хостове, хакерите могат да се свържат с файлови сървъри, все едно са локални клиенти. В идеалния случай NetBIOS не трябва да се пуска на външен интерфейс (мрежова карта, която има външен IP).

- *POP*. POP протоколът може да бъде заплаха за отдалечени клиенти, на които им е необходим достъп до електронната им поща. Причината е, че POP използва пароли в чист текст, за да позволи достъп до e-mail акаунти. Ако хакери намерят този порт отворен, те лесно могат да послушат пароли по мрежата без знанието на потребителя.
- *NFS*. Ако портът за NetBIOS сесия е оставен отворен за Интернет на UNIX хостове, хакерите могат да се свържат с файловите сървъри все едно са локални клиенти.

Има два типа защитни стени, филтриращи пакети - stateless защитни стени и stateful защитни стени. Ще разгледаме всеки от тези типове поотделно.

Stateless защитни стени, филтриращи пакети

Оригиналните филтри на пакети се наричат stateless филтри на пакети, защото те не могат да запазват информация за установена сесия между два хоста на частни и публични мрежи. Тези филтри не могат да определят дали между хостовете е установена обратната сокетна връзка. Например, хост в частна мрежа изпраща заявка за достъп до сайт с адрес 10.0.0.1 през порт 80. Пакетите на заявката, доставени до хоста на публичната мрежа, съдържат още информация за обратния сокет (IP адреси и номер на порт), на който хостът на частната мрежа иска да слуша. Един stateless филтър на пакети не може да запази тази информация. Например, не е възможно да се позволи на HTTP отговори да преминат през защитната стена само в отговор на HTTP заявки.

Stateful защитни стени, филтриращи пакети

Stateful филтри на пакети запазват състоянието на връзки чрез запис на информацията за установяване на сесия между два хоста по мрежите. На базата на тази информация филтрите решават дали пакетите върнати от публична мрежа са от доверени хостове.

Stateful защитните стени са полезни, когато трябва да се филтрира безвръзков трафик като например DNS сървъри, които се базират на UDP. Stateful защитните стени проверяват отговора на заявки чрез проверка на пакетите, чакащи отговор в таблицата на състоянията.

Stateful защитната стена, филтрираща пакети, проверява мрежовия трафик, минаващ през нея. Този тип защитни стени имат възможност да проверяват съдържанието на пакета чрез позволяване на определени типове команди в приложение, докато забраняват други команди. Например, stateful защитна стена, филтрираща пакети, позволява FTP командата GET и от друга страна забранява командата PUT.

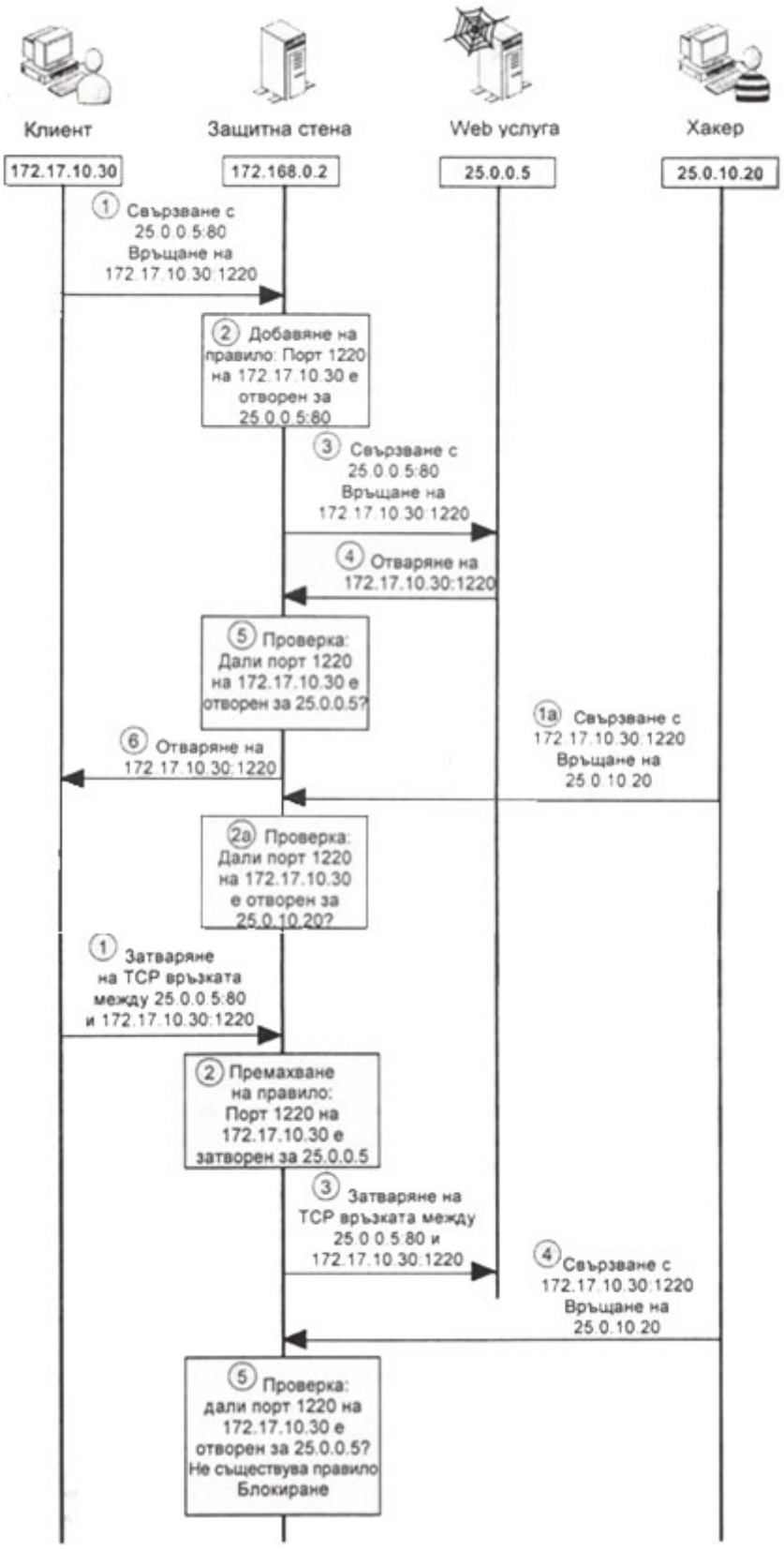
Една stateful защитна стена, филтрираща пакети, също може да подсигурява stateless протоколи. Например, в случая на приложенията, базирани на протокола UDP (DNS, RPC, NFS), със статично филтриране на пакети е трудно пакетите от данни да бъдат филтрирани, защото няма концепция за заявка и отговор. Тези приложения са наречени “stateless” приложения. Затова в случая на статичното филтриране на пакети е най-добре да се забранят UDP-базираните пакети. Със stateful филтрирането на пакети, UDP-базираните приложения могат да се подсигурят чрез създаване на виртуални връзки по и над UDP връзките.

Всяка заявка за UDP пакет, на която е позволено да премине през защитната стена, се записва в таблицата на състоянията на защитната стена. UDP пакетите, пътуващи в обратната посока, се проверяват с тези, чакащи отговор в таблицата на състоянията. Пакет, който е автентичен отговор на пакет-заявка, преминава, а всички останали се игнорират. Ако в определен период от време не пристигне отговор, връзката се прекъсва. По този начин могат да се подсигурят дори и UDP приложения.

Stateful защитните стени, филтриращи пакети, не позволяват през тях да преминават никакви услуги освен тези, които са програмирани да пропускат, и връзките, които поддържат в таблицата си със състояния.

Когато хост в частна мрежа иска да се свърже с хост в публична мрежа, заедно с пакета за синхронизация на връзката (SYN пакет) той изпраща сокета (който е IP адрес и порт), на който очаква да получи отговор. Когато SYN пакетът се рутира през stateful защитна стена, филтрираща пакети, тя прави запис на сесията в таблицата си със състояния. Сесийният запис съдържа подробности, като например получаващия сокет (IP адрес и номер на порт) и сокет на източника (IP адрес и номер на порт). Когато хостът в публичната мрежа изпрати обратно отговор, филтърът проверява записа в таблицата си, за да потвърди информацията за източника на пакета и получаващия го сокет. Ако не намери запис, пакетът се игнорира. На фигура

25 е илюстрирано как stateful защитна стена, филтрираща пакети, позволява на данни да преминат през нея.



фигура 25 *Stateful защитна стена, филтрираща пакети*

Когато се затвори TCP сесия или ако не се получат TCP пакети за затваряне на сесия след период на закъснение, stateful защитната стена, филтрираща пакети, премахва записите за TCP сесията от таблицата си. Това подсигурява, че хакери няма да използват информацията за изпуснатите връзки, за да създадат отново връзки с вътрешната мрежа.

Ограничения на филтрите на пакети

От споменатото по-рано става ясно, че има ограничения на защитните стени, филтриращи пакети. Поради това те не се използват често. Едно от тези ограничения е, че филтрите на пакети не могат да проверяват съдържанието на пакетите за наличие злонамерени данни, преди да ги предадат на вътрешната мрежа. Те разчитат на информацията в заглавието, за да вземат решения за преминаване и отхвърляне на пакетите от данни. Заради това ограничение филтрите на пакети сами по себе си не представляват ефективна мярка за сигурност на мрежи. Те трябва да се комбинират с прокси сървъри на ниво приложения или кръгово разпределени защитни стени, за да предоставят ефективна сигурност.

Защитни стени, преобразуващи мрежови адреси (Network Address Translation, NAT)

NAT (Network address Translation - RFC 3022) е подход, осигуряващ преобразуване на локални в един или няколко глобални IP адреса. Целта е представяне на локалните хостове в Интернет пространството чрез ограничено използване на глобални IP адреси. Съществуват три основни реализации на този подход, поддържани от маршрутизиращите устройства:

- статична – съпоставя на всеки вътрешен адрес различен глобален адрес;
- динамична – обвързва група от локални адреса с един или няколко глобални адреса. За целта, в NAT таблиците на маршрутизаторите, където се асоциират локален с глобален IP адрес, се добавя и порт, идентифициращ комуникационния процес;
- смесена – обединява изброените преди това реализации.

Ценната способност на NAT да осигурява сигурност доказва, че NAT е средство за ефективно скриване на вътрешните хостове.

локален адрес	глобален адрес
192.168.0.5	87.97.197.51
192.168.0.15	87.97.197.52
...	...

таблица 2 *Статично разпределение*

локален адрес	порт	глобален адрес	порт
192.168.0.5	80	87.97.197.51	6880
192.168.0.15	3389	87.97.197.51	6689
...	...	...	...

таблица 3 *Динамично разпределение*

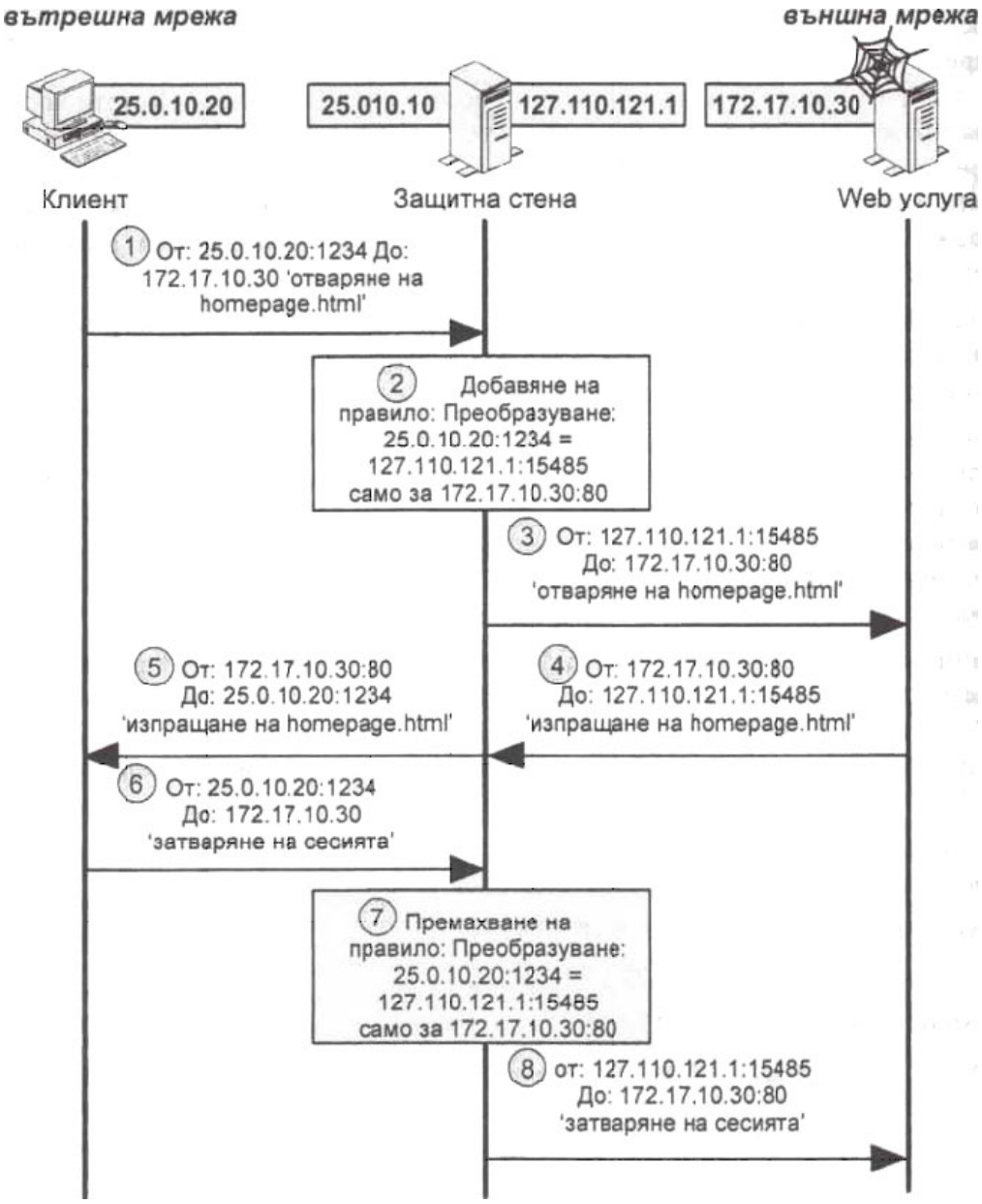
Една защитна стена изпълнява NAT функции чрез поддържане на таблица за преобразуване, която съдържа асоциирането (mapping) на вътрешни сокети (IP адреси и номера на портове на хостове във вътрешната мрежа) с външни сокети от защитна стена (IP адрес и номер на порт на защитна стена) (таблица 3). Когато хост в частна мрежа иска да установи връзка с хост в публична мрежа, защитната стена заменя вътрешния сокет с външен сокет и прави запис в таблицата си за преобразуване. Този запис показва същинския вътрешен сокет (сокета на хоста в частната мрежа), получаващия сокет (сокета на хоста в публичната мрежа) и външния сокет на защитна стена.

След като създаде запис в таблицата си, защитната стена изпраща заявка до външния хост от името на вътрешния клиент. За външния клиент изглежда, че заявката идва от друг компютър в Интернет.

В отговор на заявката, когато хостът в публичната мрежа изпраща пакетите от данни обратно на защитната стена, тя извършва обратния процес на преобразуване - опитва се да свърже сокета на външния хост със записите, които е направила в таблицата за преобразуване. Ако не намери запис за външния хост или ако IP адресът на източника (хоста в частната мрежа) е различен от адреса, който защитната стена очаква да види, пакетът се игнорира.

NAT решава няколко проблема, свързани с директните Интернет връзки, осъществени чрез защитни стени, филтриращи пакети. Но тъй като

работи на транспортния слой, NAT не ограничава напълно потока от злонамерени пакети от данни. Причината е в това, че NAT не може да проверява съдържанието на пакетите от данни, които препраща. Възможно е за протоколите от висок слой да експлоатират слабостите в трафика от висок слой. Хакерите могат да разгърнат мрежови контроли, за да шпионират трафика, излизащ от защитната стена, за да разберат има ли преобразуване на адреси. След като получи такава информация, хакерът може да превземе TCP сесии или да подправя IP адреси от защитната стена. За предотвратяване на това, е необходимо обединяване на защитните стени с прокси услуги, които работят на приложния слой.



фигура 26 *Защитна стена с възможност за преобразуване на мрежови адреси*

Circuit relay защитни стени

Circuit relay, също наречено circuit-level gateway, е вид circuit-level gateway методология на защитна стена, в която връзките на данните се потвърждават, преди данните да се обменят в действителност. Това означава, че защитната стена не само изпълнява функцията предаване/отказване на пакети от данни, но също и определя дали връзката между двата клиента е валидна според конфигурираните правила. Една защитна стена може да потвърди връзка според следното:

- Целеви IP адрес и/или порт;
- IP адрес и/или порт на източника
- Време от денонощието;
- Протокол;
- Потребител;
- Парола и др.

След потвърждаване на връзката, защитната стена отваря сесия и позволява преминаването на трафик. Но на няколко пъти защитната стена също поставя ограничения на времето, за което данните преминават.

Всяка сесия на обмен на данни се проверява за автентичност и се наблюдава и целият трафик може да преминава през времето, в което връзката е отворена. Едно предимство на една circuit relay защитна стена е, че тя покрива ограниченията на ненадеждния UDP протокол, в който адресът на източника не се потвърждава като функция на протокола.

Circuit-level защитната стена работи на ниво транспортен слой. Това понякога се превръща в недостатък за circuit-level защитната стена, тъй като може да изисква значителна промяна на програмирането, което обикновено предоставят транспортните функции, като например Winsock.

Прокси защитни стени за приложения

Думата прокси (proxy) в действителност означава едно нещо, заместващо друго. Прокси сървърите заместват директната комуникационна връзка между клиент и сървър с техните услуги. Като NAT, прокси сървърите скриват клиента от сървъра, без да нарушават комуникационната връзка помежду им. Но не това е причината, поради която прокси сървърите са първоначално проектирани. Прокси сървърите

оригинално са разработени, за да “складират” (кешират) страници. “Складиране” на Web страница означава съхраняване на нейно копие на сървъра. В ранните дни на Интернет понеже скоростта е била малка и Web страниците са били статични, повторният достъп до сайтове от много потребители правил трафика по-бавен. Затова прокси сървърите са се използвали за “складиране” на WEB страници, така че организациите да могат да елиминират повторния достъп до една и съща Web страница.

Но Интернет се разшири, скоростта се подобри, Web страниците станаха динамични и това доведе до по-малката ефективност на “складирането”. Web страниците започнаха да се променят с такава скорост, че “складирането” им вече не беше важно. Но процесът на “складиране” подчерта една по-светла страна на прокси сървърите: те могат да се използват, за да скриват хостове на мрежа зад единствена машина, те могат да филтрират URL адреси и могат да потвърждават съдържанието на пакетите от данни, които се предават. Прокси сървърите се развиха от обикновени машини за “складиране” до защитни стени.

Когато хост във вътрешна мрежа опита да се свърже с Web сайт в Интернет прокси сървърът получава заявка от хоста. Ако прокси сървърът функционира още като сървър за “складиране”, той търси заявената Web страница в паметта си. Ако Web страницата съществува в кеш паметта, той изпраща обратно страницата на хоста. Но ако Web страницата не съществува, той препраща заявката от името на хоста.

Как клиентите на мрежи взаимодействат с прокси сървъра? Клиентите имат достъп до Web страници през браузъри. Сега браузърите са настроени с адреса на прокси сървъра. Това означава, че когато клиент изпрати заявка през браузъра, браузърът автоматично изпраща всички заявки за Web страници на прокси сървъра, вместо да анализира IP адреса и да обработва заявката директно.

Не е необходимо прокси сървърът да е стартиран в защитна стена. Всеки сървър, намиращ се във или извън мрежа, може да изпълнява ролята на прокси. И двата вида - защитна стена без прокси услуги и самостоятелен прокси сървър - не могат да предоставят сигурни услуги. Един прокси сървър трябва да има някакъв вид филтриране на пакети, за да се предпази от мрежови атаки, като например атаки на услуги. По подобен начин една

защитна стена трябва да изпълнява прокси услуги, ако иска да предостави истински елементи на сигурност.

Някои функции на прокси защитна стена:

- **Прокси защитна стена с възможност за филтриране и скриване на IP**. Тези защитни стени могат блокират директни външни опити за връзка към отдалечени хостове. Тогава прокси защитната стена се свързва с отдалечения сървър и дава заявка за данни от името на клиента чрез скриване на IP (NAT функционалност).
- **Прокси защитни стени с филтриране на ниво приложения за определено съдържание**. Някои прокси защитни стени могат да се настроят с правила да проверяват съдържанието на HTML страници, отнасящи се до Java или ActiveX-вградени аплети, и да игнорират тези пакети. Това предотвратява изпълнението на аплетите и така се избягва случайно сваляне на вируси или троянски коне.

Портал за приложения

Портал за приложения (application gateway) е вид прокси защитна стена за приложения, която прави стъпка напред в контролирането на трафика по мрежата. Тя действа като прокси за приложения чрез изпълняване на целия обмен на данни с отдалечената система от тяхно име.

Един портал за приложения взема решения за пускане или отказване на базата на специфични правила като например разрешаване на някои команди на определен сървър (но не и други), ограничаване на файловия достъп до определени видове и промяна на правилата в зависимост от автентичните потребители. Този тип защитна стена също документира подробности за трафика и следи събития на системата на хоста. Той може също да бъде програмиран за звукови аларми или да известява оператора при дефинирани условия.

Порталите на ниво приложения основно се смятат за най-сигурният тип защитни стени. Те нормално се пускат на отделен компютър по мрежата, чиято основна функция е да предоставя прокси услуга.

Недостатък на портала за приложения е, че инсталирането му е много сложно, затова може да изисква особено внимание за индивидуалните приложения, които използват портала.

Предпазни мерки при реализиране на прокси защитни стени за приложения

Прокситата могат да работят само за определени приложения. Например нужда от отделен модул за HTTP услуги, друг модул за FTP и отделен модул за telnet. С развитието на тези протоколи модулите на прокситата трябва да се подновяват.

Освен това някои протоколи са или частни, или са толкова редки, че нямат достъпни съответстващи сигурни прокси. За такива приложения протоколите трябва да комуникират през защитни стени от мрежовия слой, или пакетите от данни трябва да се изпращат през основни TCP прокси, които възстановяват съдържанието и просто предават пакетите. Пример за общо прокси е прокси то SOCKS.

Поради тези недостатъци е препоръчително да не се използва прокси сървър за всички протоколи на приложения. Освен това е желателно да се използват прокси та от високо ниво, които могат да преглеждат изпълнимо съдържание, като например ActiveX и Java, в Web страници.

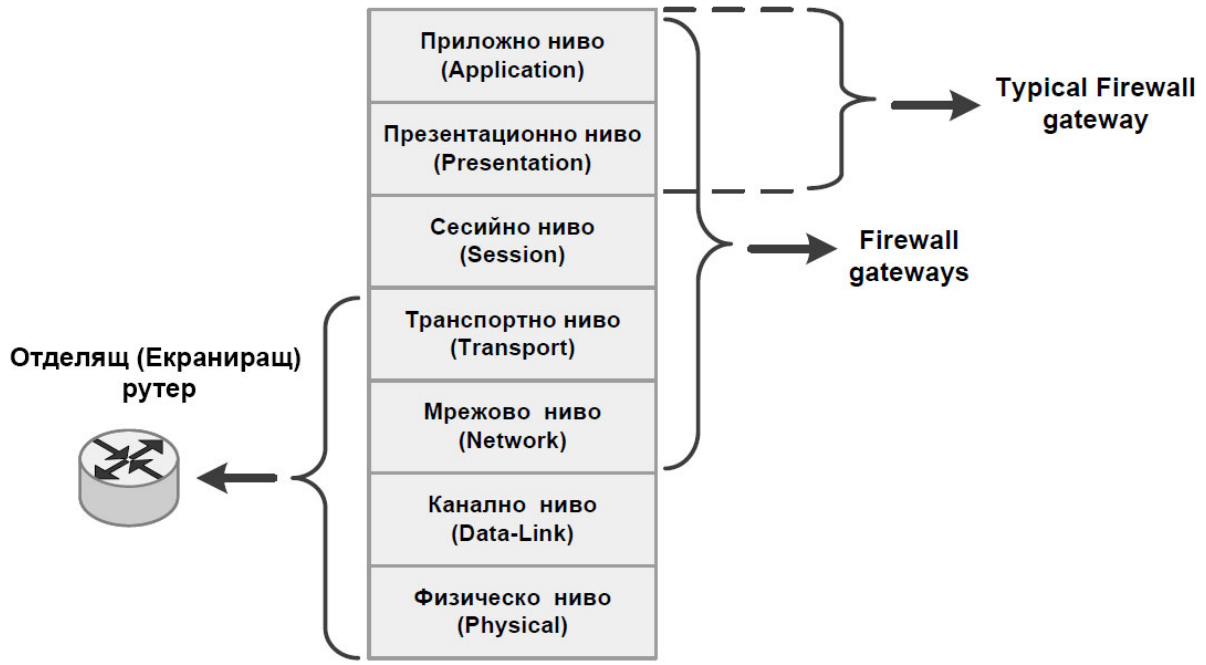
Планове на защитни стени

Инсталирането на защитна стена ограничава потока на трафика от външната мрежа (зона на риск) към частната мрежа (зона на сигурност). Какво става, ако е необходимо да се предоставят публични услуги? Възможно ли една частна мрежа да е подсигурена и да предоставя публични услуги на клиентите си и едновременно да защитава собствената си мрежа? Може да има няколко отговора на този въпрос. Но кой отговор е правилен зависи от вида сигурност, който една организация търси, и нивото на услуги, които иска да предостави клиентите си.

Разположението на защитните средства съгласно OSI модела може да бъде представено със схемата от фигура 27.

Отделящите рутери филтрират пакетите основно на мрежово и транспортно ниво, докато защитните стени функционират аналогично на гейтуей (gateways). Съществуват варианти на екраниращите рутери, които

могат да извършват филтрация на канален и физически слой. От своя страна съвременните защитни стени се произвеждат с възможности за филтриране на всички седем нива на OSI модела, поради което границата между тях и защитните рутери се размива.



фигура 27 *Разположение на защитните средства съгласно OSI модела*

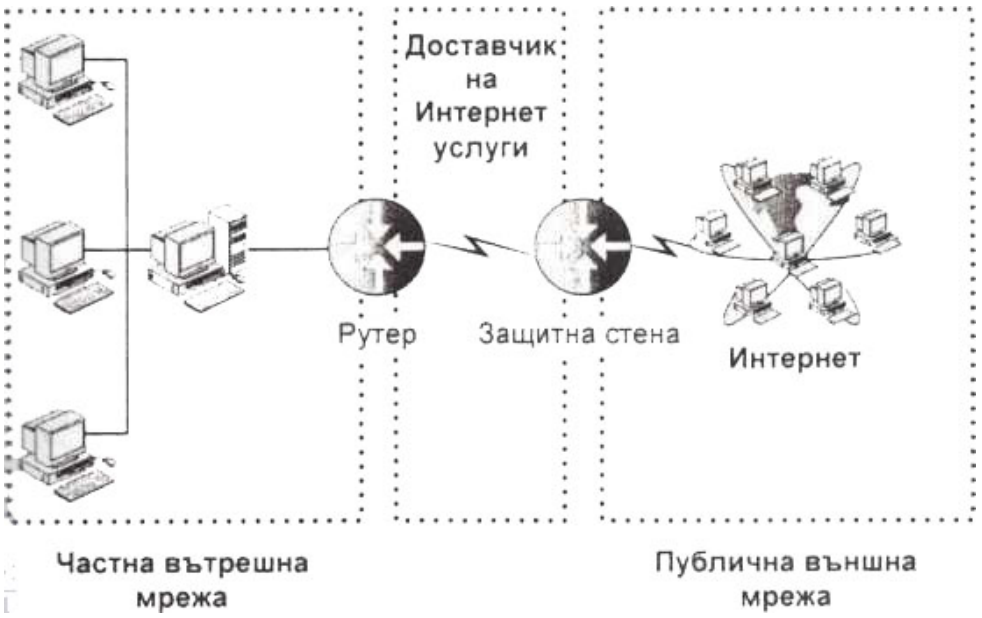
Посредством анализа на пакети, наложен върху сегментите в транспортния слой, мрежовите рутери могат да определят и възможността за съществуване на услугите (използващи този слой) в дадена крайна система. На транспортното ниво (TCP, UDP) сканирането е свързано с определяне на портовете, на които се търсят съответните услуги (Service Access Point). Транспортното ниво е отговорно за създаване на логическите (виртуални) канали.

Има няколко плана, които организациите използват, за да защитят мрежите си. Ето няколко от най-използваните планове:

- ISP филтрирани пакетни услуги;
- Единствена защитна стена;
- Демилитаризирана зона (DMZ);
- Защитна стена с отделен хост;
- Защитна стена с отделена подмрежа.

Пакетни услуги, филтрирани от ISP

Повечето доставчици на Интернет предлагат възможността за филтриране на пакети като допълнителна услуга за клиентите си. Доставчиците настройват техни собствени защитни стени да филтрират трафика до и от частни мрежи. В допълнение към предоставянето на филтриране на пакети, някои доставчици също предоставят услугите на прокси сървъри и преобразуване на мрежови адреси. На фигура 28 е показана организацията на защитна стена, филтрираща пакети, на Интернет доставчик.



фигура 28 *Защитна стена на интернет доставчик*

Единственото предимство на вземането на защитна стена, филтрираща пакети, от Интернет доставчик е, че организацията не трябва да правят големи разходи по нейното инсталиране. Но подобно инсталиране носи рискове за сигурността. Дори и Интернет доставчикът да осигурява пълно решение на сигурността, една организация все още не е в безопасност, защото нейната мрежа е в ръцете на друга организация. Никога не може да има увереност за намеренията на другата организация и нейните служители, особено в случай на някакъв спор между организацията и Интернет доставчика.

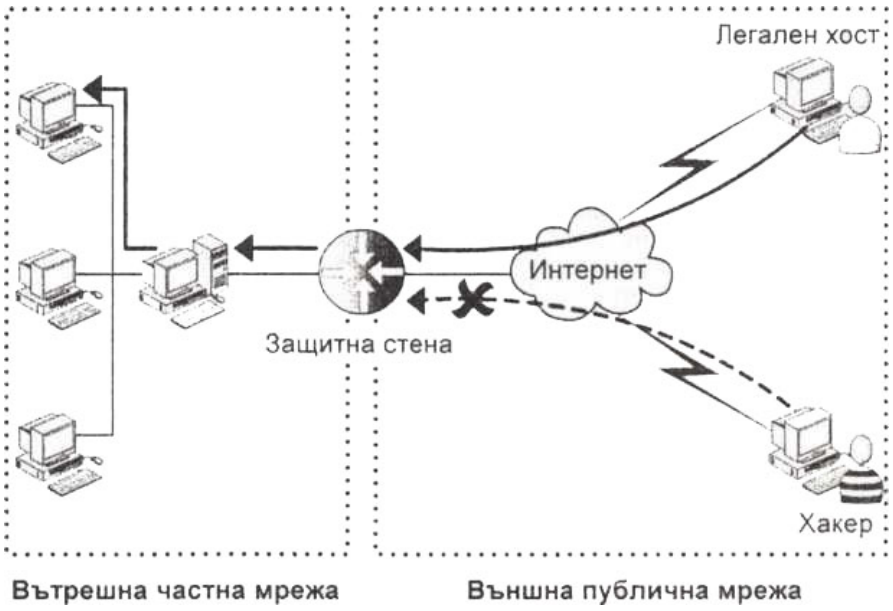
Освен тези рискове, конфигурацията е друг административен nроблем. Трудно е да се разчита на отдела за работа с клиенти на ISP за промяна на правилата на сигурността и за повторно конфигуриране на защитната стена.

В допълнение, частната мрежа е още уязвима към другите абонати на ISP, които обикновено са в същата защитна стена.

План с единствена защитна стена

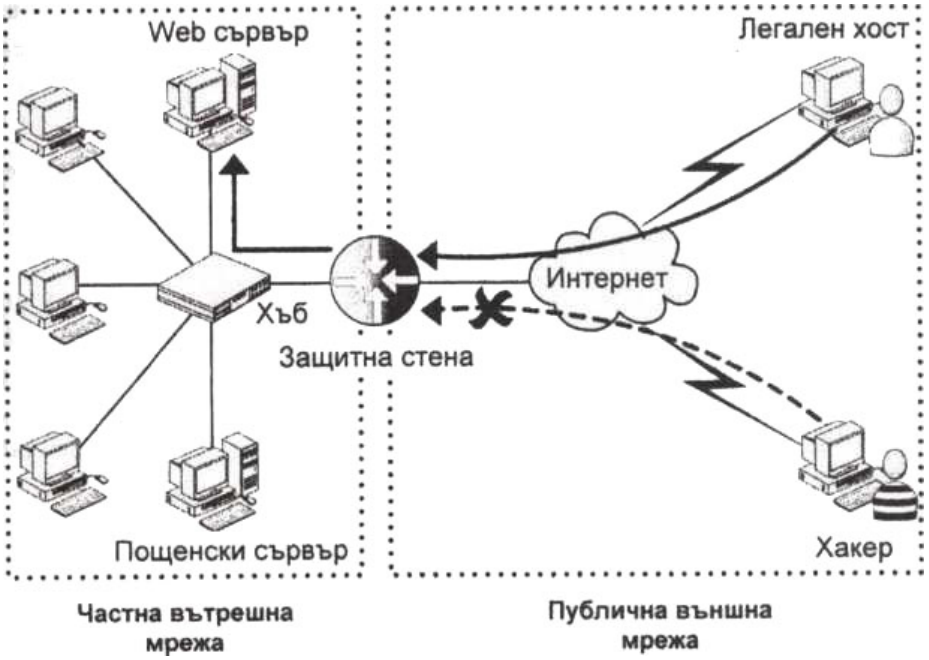
Единствена защитна стена е най-основният план на защита. В този тип план двете мрежи просто са отделени една от друга чрез защитна стена помежду им.

Единствената защитна стена се конфигурира не само да подсигурява частната мрежа от Интернет, но и да позволява на потребителите на частната мрежа достъп до Интернет. Тъй като има само една защитна стена и само една връзка с Интернет, има една точка на контрол и управление в подобен план.



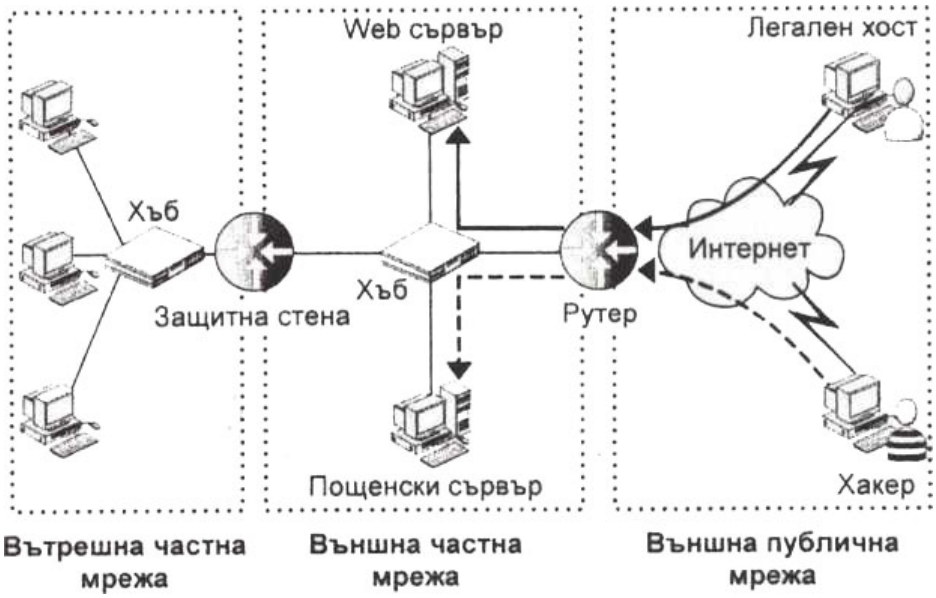
фигура 29 *План с единствена защитна стена*

Всичко е наред, докато организацията иска да отдели мрежата си от външната публична мрежа, която е Интернет. Но може да е необходимо организацията да предостави публични услуги, като Web и FTP, на клиентите или може да възнамерява да работи с e-mail сървъри. В такива ситуации за организациите има няколко възможности. Първо организацията може да постави публичните си сървъри зад защитна стена и така да отвори връзка до външната мрежа за публичните сървъри на други организации. Този метод още се нарича „портална двудомна защитна стена”. Защитната стена има две карти за интерфейс, една за доверената мрежа и другата - за недоверената мрежа.



фигура 30 *Единствена защитна стена с достъпни сървъри*

Във втората възможност организацията може да постави публичните си сървъри извън обсега на защитната стена, така че да са открити за публичната мрежа. На фигура 31 е показано разположението на публичните сървъри на организация след защитната стена.



фигура 31 *Единствена защитна стена с открити сървъри*

Този план, в който публичните сървъри на организации са зад защитната или след защитната стена и са открити за света, се нарича още ненадежден план на хостове. Тук ненадежден се отнася за публичните

сървъри, защото те могат да позволят нахлуване в частната мрежа на организация.

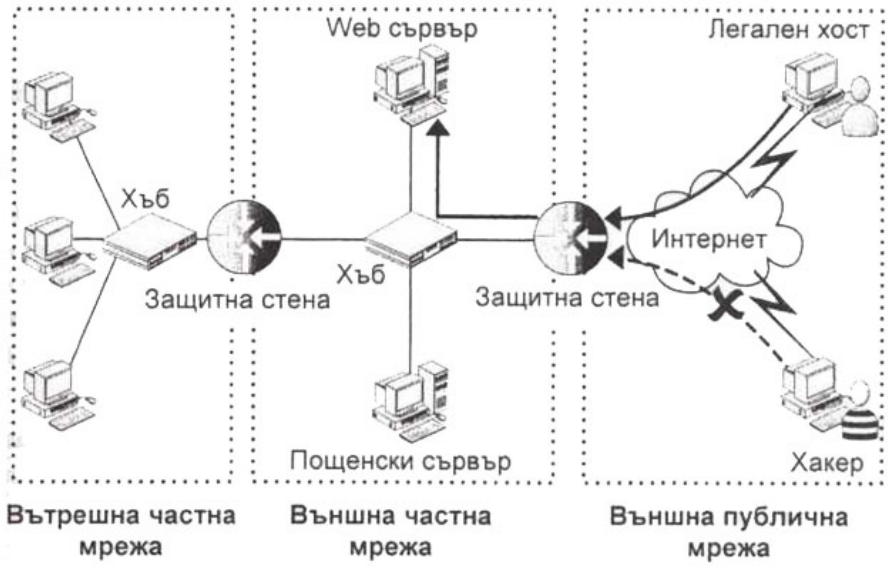
Проблемът с поставянето на публичните сървъри на организация извън защитната стена е, че те са отворени за неограничени хакерски опити. Този тип план е подходящ само ако сървърите не съдържат много полезна информация.

Проблемът с отварянето на връзка през защитна стена, за да може външната мрежа да получи достъп до публичните сървъри е, че всеки пакет може да влезе в частната мрежа, ако при проверка отговаря на използваната защитна стена. Това означава, че хакери, които могат да експлоатират софтуер за услуги от високо ниво, могат да получат достъп до частната мрежа.

План с демилитаризирана зона

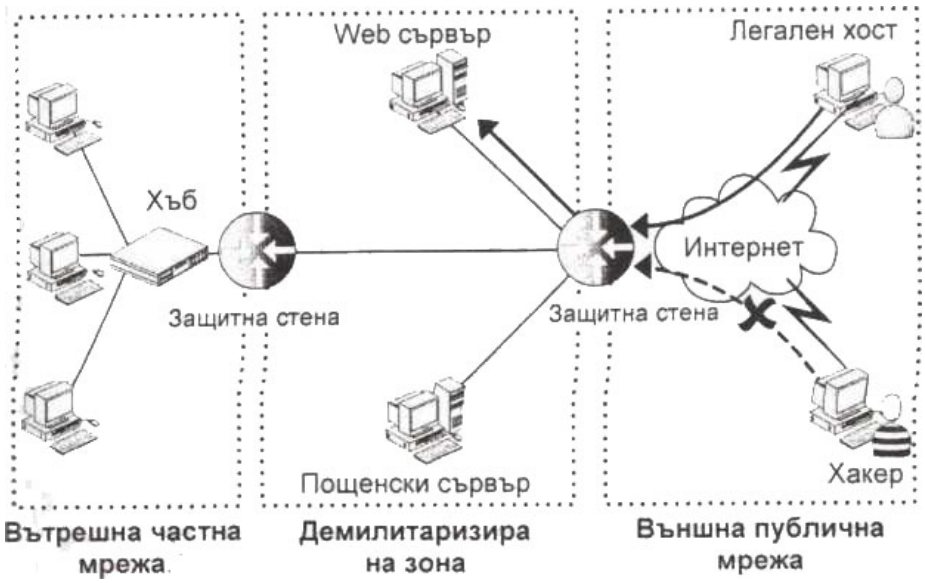
Дискутираните планове в предходната секция включват потенциален риск за вътрешната мрежа. Рискът от открити публични сървъри може да се намали с използване на две защитни стени и две нива на сигурност. В такова структуриране първата защитна стена се поставя при Интернет връзката, а публичните сървъри на организацията се поставят зад нея. Това предоставя сигурност, като позволява и опити на клиентите във външната мрежа за връзка до публичните сървъри на организацията.

Втората защитна стена се поставя между частната мрежа и публичните сървъри на организацията. Тази структура предоставя сигурност на частната мрежа, защото сега е свързана с всички външни връзки и нейната структура е напълно скрита от външния свят. Областта между двете защитни стени се нарича демилитаризирана зона (Demilitarized Zone, DMZ) (фигура 32). В плана с демилитаризирана зона (DMZ плана), когато хакер се опита да нахлуе в мрежата на организация, той е блокиран на двете нива. Ако успее да премине през първата преграда, той бива спрян на второто ниво на сигурност. DMZ планът се определя като най-безопасен механизъм за сигурност на организации.



фигура 32 *Демилитаризирана зона (Demilitarized Zone, DMZ)*

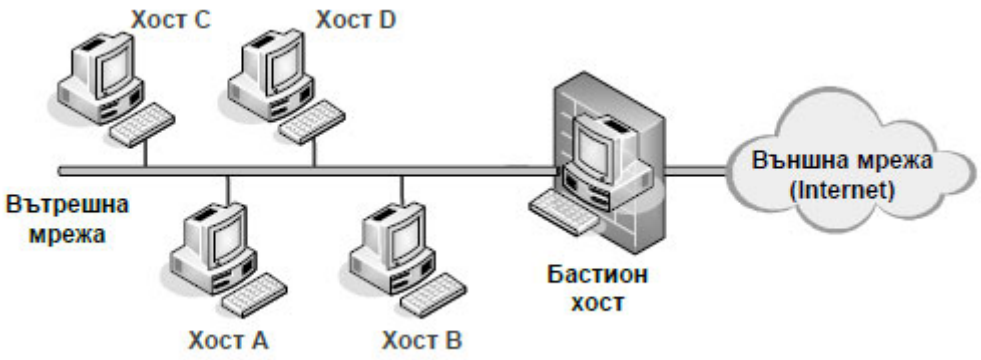
Защитните стени позволяват още използването на план с виртуална демилитаризирана зона. На фигура 33 е илюстрирана виртуална DMZ, създадена със структурата на единствена защитна стена. В тази структура защитната стена съдържа три интерфейса, свързани с външната мрежа (т.е. външната мрежа, вътрешната мрежа и мрежата на публичните сървъри), с три различни политики за сигурност. Политиките за сигурност могат да се настроят да блокират опити за връзка до вътрешната ви мрежа, но да заобикалят мрежата ви от публични сървъри. По този начин използвате двете защитни стени през единствен продукт. Този тип защитна стена се нарича “тридомна защитна стена”.



фигура 33 *Тридомна защитна стена*

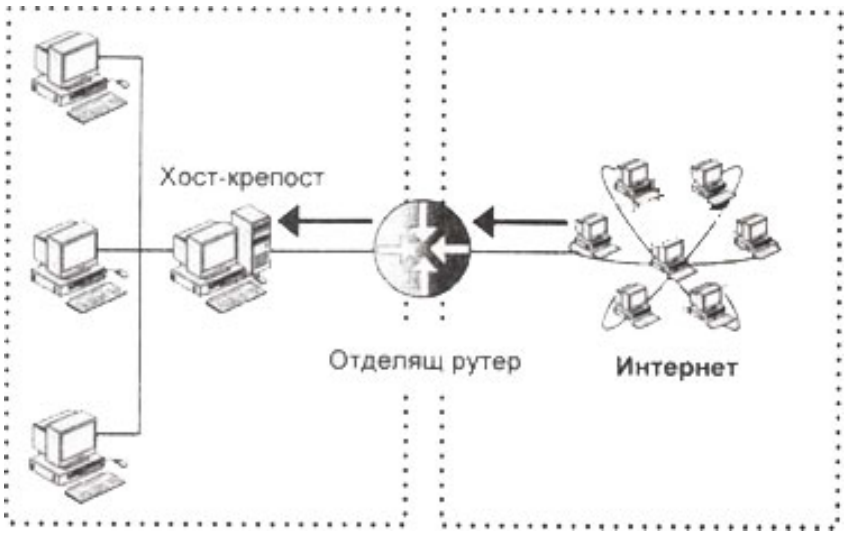
Защитна стена с отделен хост (Bastion host)

Бастион-хостът е защитен хост, критичен по отношение на мрежовата сигурност. Бастион-хостът (хост-крепост) е компютър, управляващ входящия и изходящия трафик в дадена мрежа с възможности за филтрация на всички нива от OSI модела (фигура 34).



фигура 34 *Bastion host*

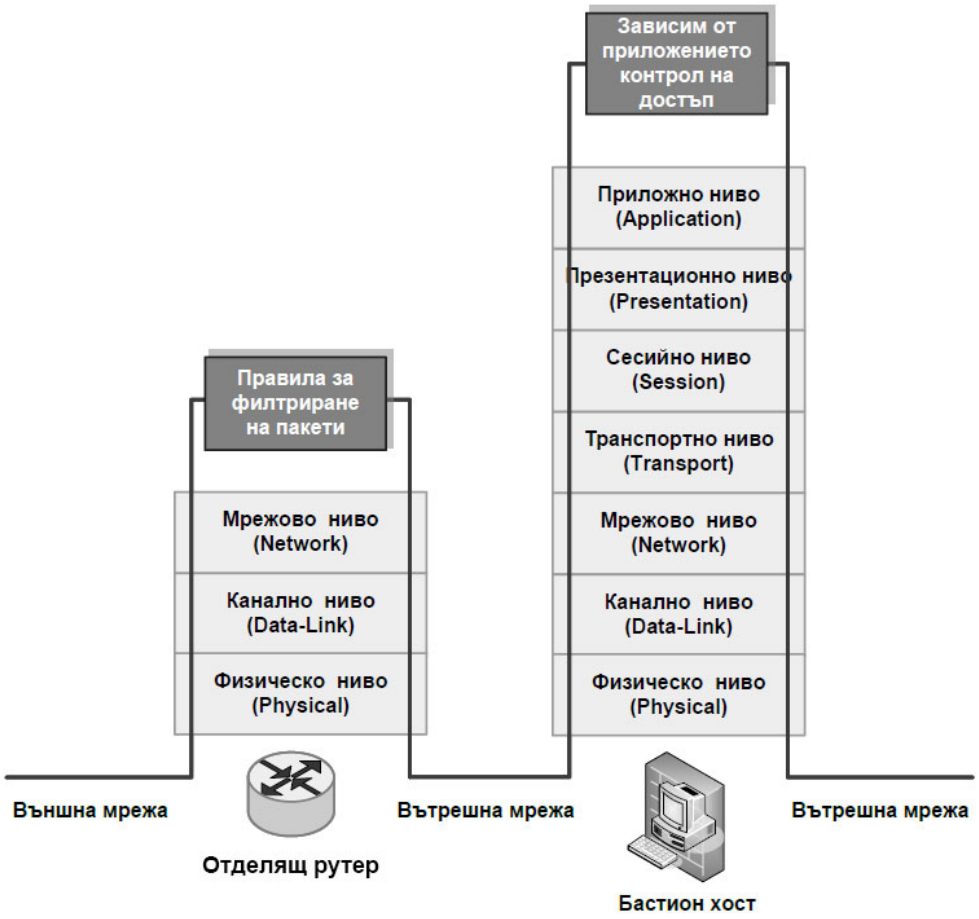
Един от плановете за защитна стена с отделен хост се състои от хост-крепост и отделящ рутер. При този план бастион-хоста се изтегля на втора линия. Отделящият рутер е първата спирка между Интернет и вътрешната мрежа (фигура 35).



фигура 35 *Хост-крепост и отделящ рутер*

Основната роля на отделящия рутер е да анализира пакетите от данни, които се опитват да се свържат с частна мрежа на базата на правила, определени в рутера. Пакетите от данни биват отказани, ако не преминат правилата на политиката за сигурност. Ако пакетите от данни преминат през

правилата, те се обработват и в хоста-крепост на частната мрежа (фигура 36). Трафикът, генериран от вътрешната към външната мрежа (outgoing traffic), може да се пренасочи директно към отделящия рутер. В този аспект на дейността си, бастион-хостът изпълнява функции на гейтуей. Затова тази схема на защита се нарича screened host gateway.



фигура 36 *Функционалност спрямо OSI модела*

План за защитна стена с отделена (екраниращa) подмрежа

Планът за защитна стена с отделена (екраниращa) подмрежа също не е нещо различно от плана с демилитаризирана зона. Както демилитаризираната зона, този план също съдържа две защитни стени. Но освен това той също съдържа хост-крепост в мрежата.

В действителност, в случая със защитна стена с отделена подмрежа, просто укрепвате машините в DMZ, като всички машини в DMZ са хостове-крепости. В общи линии подсигурявате компютрите в DMZ.

фигура 37 *План за защитна стена с отделена (екраниращa) подмрежа*

На фигура 37 хостът-крепост и някои други хост компютри формират подмрежа. Подмрежата е теоретически част от частна мрежа, състояща се от компютри, които имат обща първоначална част на IP адресите си. Подмрежата се намира между два отделящи рутера.

Един отделящ рутер се намира между Интернет и отделената подмрежа. Вторият отделящ рутер се намира между отделената подмрежа и частната мрежа. Подмрежата се нарича отделена подмрежа, защото трафикът се отделя от двата края на рутерите. Отделената подмрежа съдържа хост-крепост, Web сървър и FTP сървър. Тази отделена подмрежа още се нарича демилитаризирана зона.

Когато клиент иска достъп до ресурси от Web сървъра на организация, той изпраща заявка през отделящия рутер до отделената подмрежа. В отделената подмрежа заявката преминава през хоста-крепост, който изпълнява прокси операции на пакетите от данни, преди накрая да отговори на хоста във външната мрежа. По подобен начин, ако хост в частната мрежа иска достъп до определени ресурси от публичните си услуги, заявката първо се отделя през рутера и след това преминава през хоста-крепост.

Защитна стена с отделена подмрежа се използва, когато се изисква високо ниво на сигурност за компютри в LAN. Освен предоставянето на сигурност, защитната стена с отделена подмрежа също скрива вътрешната мрежа на организацията от публичната мрежа.

Избор на решение със защитна стена

На пазара има няколко достъпни решения със защитна стена. Изборът между тези решения е трудна и аналитична работа. Всъщност, изборът и осъществяването на решение за сигурност е жизненоважна и скъпа работа. Една организация може да избере сама - да подбере, инсталира и управлява самостоятелно решение със защитна стена или може да прехвърли работата на експерти в областта на решенията със защитни стени. Във всеки случай трябва да се разгледат определени фактори, преди да се избере подходящото решение със защитна стена:

- Определяне на уязвимите точки в мрежата;
- Оценка на стойността за изграждането на защитна стена;
- Установяване на необходимите условия за инсталиране на защитна стена;
- Избиране на подходящ тип решение със защитна стена;
- Избиране на подходящ план на защитна стена;
- Оценяване на възможностите на защитната стена;
- Сравняване на изискванията с възможностите;
- Оценяване на решението със защитна стена.

Определяне на уязвимите точки в мрежата

Първата стъпка при избора на подходящо решение със защитна стена е определяне на уязвимите точки в мрежа, които могат да представляват заплаха за сигурността на организацията. Уязвимите точки не включват само мрежови ресурси като оборудване на портал, сървъри, работни станции и мрежово оборудване, а също и услуги като Интернет, FTP, e-mail и важни бизнес приложения. Един администратор на организация трябва да определи често срещаните заплахи чрез изучаване на тенденциите на атаките, случващи се на пазара, и анализиране миналия опит на организацията с внимание към всички атаки, от които тя е пострадала. Освен това администраторът трябва да определи специфични приложения, използвани в организацията, които имат своите изисквания за сигурност. Например приложението, което отделът за човешки ресурси използва, за да скрие критична информация от служителите, има специфични изисквания за сигурност.

Определянето на уязвимите точки на организация дава на администратора широка перспектива. Администраторът може да се съсредоточи над основните области на сигурността на организацията. Освен това определянето на тези уязвими точки помага на администратора да измисли тяхната замяна и да предостави приложимо решение за сигурност.

Оценяване на стойността за реализиране на защитна стена

При избора на сигурно решение за мрежата си, организацията трябва да оцени стойността на осъществяването на това решение спрямо ползата, която може да се получи от него. Оценяването на тази стойност е като оценката на стойността на всеки друг компонент. То включва основно изчисляване на цената на покупката и поддръжката на решението.

Цената на осъществяването на защитна стена включва следното:

- *Покупка*. Цената на покупката на защитна стена е пазарната цена на продукта. Преди да купи решение със защитна стена, администраторът трябва да вземе оферти от няколко продавача и да потърси в Интернет различни достъпни решения със защитна стена. Има продукти за защитни стени, достъпни като цялостни хардуерни и софтуерни решения. Тези решения са скъпи, но лесни за осъществяване. Свободните за сваляне защитни стени са също достъпни, но те може да не предоставят желаното решение, тъй като са общи.
- *Инсталация*. Цената на инсталацията включва цената, която производителят взема, за да структурира (инсталира и конфигурира) продукта и специфичното за продукта обучение, което персоналът може да изисква като предпазна мярка, за да поправя проблеми в защитната стена.
- *Администрация*. Цената на администрацията включва цените, които се плащат при промяна на политиката за сигурност на организацията на базата на текущото естество на заплахите, цената, плащана при интерпретиране на документираната информация, и цената за последващи коригиращи действия.
- *Поддръжка*. Повечето големи организации препращат задачите по поддръжка на защитната стена на фирми за мрежова сигурност. Тези фирми за сигурност вземат пари от организацията под формата на годишни или месечни такси за

услугите си, което представлява цената на поддържане на защитната стена. Задачите по поддръжката могат да варират от поправка и настройки в политиките на защитната стена до фиксиране на грешки и решаване на проблеми всекидневно.

Установяване на необходимите условия за инсталиране на защитна стена

Следващата стъпка е да се определят основните изисквания, съществени за инсталирането на защитна стена. Една организация трябва ясно да разбира следните въпроси, защото те стоят в основата на структурирането и инсталирането на защитна стена:

- Организациите трябва ясно да разбират нивото на сигурност, което искат да предоставят на мрежата си.
- Трябва да се подготви основна политика за сигурност на защитната стена и тя трябва ясно да дефинира областта на достъп на потребителите до мрежовите услуги. Политиката трябва също да планира правилата за разрешаване на услуги като например e-mail и FTP сървър.
- Трябва да се подготви подробен отчет за документиране на събитията и документите за проверки, защото те улесняват дефинирането на политика за сигурност.
- Изискванията на мрежата трябва ясно да бъдат изложени. Например, дали мрежата изисква преобразуване на мрежови адреси и рутиращи функции? Ако мрежата изисква тази функционалност, проектирането ѝ трябва да се подготви съобразно изискванията.
- Трябва да се подготви документ, който ясно изброява операционната система и хардуерните и софтуерните ресурси, които организацията използва в момента. Това улеснява мрежовите администратори при решението дали някой софтуер или хардуер трябва да се поднови преди инсталирането на защитната стена.
- Трябва да се подготви подробен анализиращ отчет, който дефинира типовете атаки и дупките, които водят до тези атаки в организацията. Този отчет трябва също да дефинира текущо използваните контроли на сигурността и причините за провала на тези контроли за противопоставяне на атаки.

Избор на подходящ тип решение със защитна стена

Една организация трябва да избере подходящото решение със защитна зависимост от необходимите условия и изисквания, които са определени, оценената стойност на продукта и неговото бързодействие. Организацията може да избере да осъществи решение със защитна стена на ниво приложения или решение със защитна стена на ниво мрежа. Следната дискусия изброява ситуациите, когато трябва да се избере защитна стена от ниво приложения или мрежа. Изборът на защитна стена на ниво приложения, е ако:

- организацията трябва да осъществи здрав инструмент за филтриране на пакети;
- високата мрежова скорост не е основният въпрос;
- мрежата на организацията има високотехнологични хардуерни системи, които могат да поддържат високите изисквания за обработка на софтуера на защитната стена;
- организацията трябва да следи и мрежово-, и приложно-базираната комуникация;
- организацията иска да включи допълнителни елементи на сигурност, като например откриване на пробиви.

Изборът на защитна стена от ниво мрежа, е ако:

- организацията иска да използва устройство за защитна стена или рутер, който е преконфигуриран със софтуер за защитна стена и лесно може да се закачи за структурата на мрежата и независимо от използваните операционни системи;
- мрежовата скорост и изпълнение са основни въпроси;
- организацията търси ефективно относно цената решение със защитна стена.
- организацията иска да осъществи елементи, като например филтриране и докладване на IP, без да инсталира допълнителни неща.

След определяне на слоя, на който защитната стена ще бъде осъществена, трябва да изберете технология за защитна стена.

- Организация може да избере да използва приложни прокси услуги, ако планира да предостави сигурност на ниво

приложения. Прокси приложенията също предоставят възможност за маскиране на IP, което скрива вътрешна мрежа от външна такава, дори могат да включват допълнителни възможности за сигурност, като например система за откриване на пробиви.

- Една организация може да избере stateful защитна стена, филтрираща пакети, ако иска пакетите от данни да се преглеждат и потвърждават, преди да влязат в мрежата.
- Една организация може да избере хибридна система, защото тя предлага и филтриране на пакети, и прокси защитна стена.

След като определи нивото на сигурност на защитната стена и типа на сигурността, организацията трябва да реши какъв ще е режимът на усвояване на решението със защитна стена. Има няколко продукта за защитна стена, достъпни за свободно сваляне, както и такива на определена цена. Патентованите защитни стени са най-често хардуерно-базирани и са подходящи за организации с обширна структура. Причината е в това, че продавачите не само инсталират решението със защитна стена в мрежа, но и предоставят поддръжка, ако има проблем. Съществуват и софтуерно-базирани патентовани защитни стени, като например Gauntlet и MS-ISA сървър.

Избор на подходящ план на защитна стена

Друго важно решение пред организациите е кой тип план на защитна стена да използват. Следват няколко ситуации, които накратко ще ви запознаят с примерите за план:

- План с единствена защитна стена е най-подходящ, когато организацията се нуждае да реализира икономично решение със защитна стена. Това означава, че организацията планира да използва само един рутер за осъществяване на функции на защитна стена, планът с единствена защитна стена е най-добро и най-ефективно решение относно цената.
- Ненадеждният хост план, при който публичните сървъри на организации са поставени зад защитната стена или след нея и са отворени за света, е подходящ, когато публичните сървъри на частна мрежа не изискват висока сигурност и не разстройват услугите на вътрешната мрежа.

- Планът с демилитаризирана зона е най-подходящ, когато една организация иска да защити публичните си сървъри, както и вътрешната си мрежа.

Оценка на възможностите на защитната стена

Няколкото възможности, които трябва да се разгледат, са мащабируемост, ниво на сложност, изпълнение и допълнителни възможности.

Мащабируемост

Една защитна стена е мащабируема, ако може да обедини други нови приложения и защитни стени. Например, дори и след структурирането на решение със защитна стена, една организация може да има нужда да поднови системата си с антивирусни обновявания (updates) или да има нужда да инсталира система за откриване на пробиви, за да подсили сигурността. Защитната стена трябва да е достатъчно гъвкава, за да обедини тези промени с минимални или никакви промени в структурата си.

Ако тя не е мащабируема, организацията може да има нужда да купи ново решение всеки път, когато иска да добави нови възможности.

Сложност

Една защитна стена не трябва да е толкова сложна, че да е трудно да се разбере, инсталира, конфигурира и поддържа. Лесни инсталационни стъпки елиминират риска от грешни конфигурационни настройки. Освен това интерфейсът на защитната стена трябва също да е прост и последователен, така че да е лесно да се администрира. Но улесняването на сложността на защитна стена не означава да я направите толкова елементарна, че всеки външен човек да може да хакне мрежата.

Бързодействие

Бързодействието на решение със защитна стена трябва да подсилва скоростта и бързодействието на цялата мрежа на организацията, вместо да я затруднява. Защитната стена трябва да бъде достатъчно сигурна, за да управлява тежък трафик на данни. Тя трябва да предоставя възможност за автентикация, за да потвърди влизането на потребители и пакети от данни в мрежата.

Освен това теоретически една защитна стена трябва да се конфигурира така, че редовно да сваля и инсталира обновявания на продукта. Тя трябва да може да “съживява” мрежата бързо в случай на срив.

Допълнителни възможности

Възможностите, дискутирани по-горе, са основни елементи, които са задължителни за всички решения със защитна стена. Но има определени допълнителни възможности, които могат да направят едно решение със защитна стена цялостно:

- Решенията със защитна стена е за предпочитане да съдържат функционалност за тунелиране, за да осъществяват решение на криптиране сайт-до-сайт.
- Защитните стени трябва да документират дейността на мрежата, така че администраторите да могат да проследят всяко събитие за деня.
- Защитните стени трябва да имат вградена висока достъпност, за да се занимават с мрежови рискове, които могат да се появят поради непредвидени крахове. Тази особеност позволява на защитните стени да предават операциите си на backup защитни стени, ако има срив.
- Някои защитни стени също предоставят механизми за хващане на нарушители. Един такъв механизъм е honeypot механизмът, който примамва нарушители с показване на данни, които не са истински.

Сравнение на изискванията с възможностите

Когато веднъж организацията е оценила възможностите на решението със защитна стена, следващата важна стъпка е да се сравнят изискванията за мрежова сигурност с възможностите на различните решения със защитна стена, достъпни на пазара. Трябва да се избере само това решение, което изпълнява повечето мрежови изисквания. Трябва да се внимава при оценяване на особеностите на защитната стена. Оферти за решение със защитна стена трябва да се вземат от различни продавачи. Освен това преди завършването на сделката организациите трябва да проверят поддръжката след продажбата, която продавачите предлагат да предоставят. Поддръжка след продажбата е жизненоважен аспект на процеса на осъществяване на защитна стена.

Оценка на решението със защитна стена

След като сте сравнили възможностите на защитната стена с изискванията за мрежова сигурност и преди накрая да изберете решение със защитна стена, е препоръчително да оцените решението със защитна стена по следните фактори:

- *Техническа поддръжка след продажбата*. Продавачите трябва да предоставят ефикасна техническа поддръжка не само по време на инсталацията на защитната стена, но и след продажбата. Това подсигурява, че редовна поддръжка е достъпна в случай на срив или когато тя се изисква.
- *Цена на поддръжката*. Поддържането на защитната стена е част от нейната обща цена. Много организации правят грешка, като не добавят цената на поддръжката в общата цена на продукта и когато се плати за поддръжка (цените обикновено са доста високи), те обвиняват продавачите, че не са ги информирали по-рано. Затова, когато подписват договор с продавач, организациите трябва да проверят периода на гаранцията и клаузата за гаранция, която им предлагат. Това ще помогне на организацията да оцени типа и размера на поддръжката, която продавачът предостави.
- *Документация на продукта*. Организациите трябва да проверят д продуктът, който искат да купят, съдържа подробна документация инсталиране, конфигуриране и поправяне на защитната стена.
- *Прототип на защитната стена*. Преди една организация да купи продукт със защитна стена, тя трябва да попита за прототип на продукта, да тества и оцени функционирането му. Това помага за оценяване слабите и силните страни на продукта. Освен това организацията получава възможност да потвърди, че решението със защитна стена предоставя цялостна сигурност на мрежата.

Лекция 5

Какво е откриването на пробиви?

Откриването на пробиви е изкуството за откриване и реагиране на компютърни атаки и злоупотреби. Неговите функции включват широко предпазване, възпиране, откриване, реагиране, оценка на щетите, очакване на атаки и поддържане на информация за пробивите.

Има различни техники за откриване на пробиви и всяка от тях осигурява различни ползи в зависимост от обкръжението. Техника, която е удачна в едно обкръжение, може да не е удачна в друго. В следствие на това, може да откриете, че се срещат няколко термина за откриването на пробиви. Най-често използваните (но не задължително най-точните) са следните;

- Intrusion detection означава откриване на неупълномощен достъп до компютърна мрежа.
- Misuse detection означава откриване на активност, съвпадаща със "злоупотреба" по зададени правила.
- Anomaly detection означава откриване на нарушения от приемливия профил на поведение.
- False-positive е аларма за някои неестествени дейности, които не са злоупотреба (фалшива тревога).
- False-negative е злоупотреба, която не се открива или алармира.

Това са специфични дефиниции, които наблягат на различни аспекти на откриването на пробиви. Най-общите дефиниции, които обясняват широките възможности на системите за откриване на пробиви, са дадени от Intrusion Detection SubGroup (IDSG) от президентския съвет National Security Telecommunications Advisory.

Пробивът (intrusion) е неупълномощен достъп до и/или активност в информационна система.

Откриването на пробив (intrusion detection) е процесът на идентифициране, че е направен опит за пробив, че пробивът се случва в момента или че вече се е случил.

Индикацията (indication) е информация, която предполага заплаха. Тя включва. специфични доказателства, че се е случил пробив, и включва интересите, намеренията и възможностите на заплахата.

Системите за откриване на пробиви (IDS) осигуряват допълнителни възможности отвъд откриването и отговарянето на злоупотреби и нарушители. Няколко от тези възможности включват поддръжка при определяне на глоба и завеждане на дело. Имало е няколко опита по-ясно и прецизно да се дефинират условия, отнасящи се до откриването на пробиви, но терминът intrusion detection остава най-общ термин за широк набор от възможности. Въпреки това, всички дефиниции са приемливи. Можете да си насочите вниманието към специфичните изисквания за откриване и реагиране, докато съвпаднат с инструмент или набор от възможности.

Как работи IDS?

Традиционно, за да откриват всеки опит за пробив или да анализират данните, администраторите ръчно наблюдават журналите (logs), генерирани от системите за сигурност (като защитна стена). В последно време процесът по наблюдение на ресурси, опити за пробив или детайли за необичайно потребление на ресурси и анализ се поемат от IDS. IDS също наблюдава използването на процесора, вход-изход от диска, паметта, потребителската активност и количеството опити за влизане чрез идентифициране. Ако записаните журнали покажат отклонение от нормалното използване, IDS реагира с аларма.

Как IDS прави това? IDS поддържа база със сигнатурни файлове, в които има шаблони на атаки.

Какво е шаблон? Всяка атака си има подпис, шаблон и поведение. Тази комбинация е позната като сигнатура. IDS разпознава атаката или пробива, като сравнява за съвпадение със сигнатурните файлове в базата.

Понякога се случва IDS и да не разпознае атаката или опита за пробив или да я вземе за нормална активност. В резултат на това се генерират грешки. Те могат да се класифицират като false-positive или false-negative.

False-positive грешка се генерира, когато IDS сметне нормална активност по мрежата за опит за пробив и генерира аларма. Ако този тип аларма продължава да се генерира, администраторите може да започнат да

не забелязват истинските аларми. В резултат на това реалните опити за пробив може да останат незабелязани.

False-negative грешка се случва, когато IDS игнорира опит за пробив и го сметне за нормално поведение на мрежата.

IDS и защитна стена

Често има объркване между функциите на IDS и защитната стена, като неправилно се интерпретират функциите на защитна стена, които всъщност са функции на IDS. Придържайки се към общото разбиране, защитните стени разпознават атаката и я блокират. Това обаче не е съвсем вярно.

Защитните стени работят, като блокират всичко и тогава програмно допускат само няколко избрани неща. В един идеален свят, всички системи и мрежи биха били блокирани и осигурени и защитна стена не би била необходима. Точната причина да използването на защитна стена е за запушване на някои дупки в сигурността.

Затова, когато е инсталирана защитна стена тя първо спира всякакви комуникации. След това администраторът добавя правила, които разрешават преминаването на специфични типове трафик и възпира всеки друг тип трафик. Например, една защитна стена, разрешаваща достъп до Интернет, блокира всички UDP и ICMP дейтаграми, спира всички приходящи TCP връзки, но разрешава изходящи TCP връзки. С други думи, защитната стена разрешава вътрешните потребители на организацията да използват Интернет, но отказва всички заявки от Интернет към вътрешната мрежа. Защитната стена не е динамична система, която може да усети, че се провежда атака.

За разлика от защитната стена, IDS е много по-динамична система. Тя има възможността да открива и разпознава атаките срещу мрежата, които защитната стена не може да види.

Представете си следния пример на пробив. Служител на организацията получава електронна поща от друг служител, казваща, че той е открил отдавна липсващия му файл. Служителят отваря пощата и щраква върху изпълнимия прикачен файл, който е архивиран. Изпълнимият документ има троянски кон, прикачен в себе си. Троянският кон, отваря връзка към хакерския компютър. Сега защитната стена, инсталирана в мрежата, не възпира хакера да използва своя атакуващ софтуер на обикновения порт 80.

Това е така, защото защитната стена е програмирана да не допуска външни връзки само през определени портове. Тя смята всяка друга външна HTTP връзка, излизаща от вътрешната мрежа към Web сървър, просто като друга връзка.

Ако е била инсталирана, IDS би могла да вдигне аларма за необичайна активност в мрежата. За пример с предходния случай, в нормални условия служителите не биха опитвали да влязат във връзка със сайта, към който троянският кон опитва да отвори връзка. Това е необичайна активност и следователно ще се генерира аларма.

IDS техники

За да открие пробиви, IDS системата трябва да е базирана на следните две техники:

- Anomaly-detection техника (откриване на аномалия);
- Misuse-detection техника.

Anomaly-Detection техника

Anomaly-detection техниката е базирана на предположението, че всички действия, несъвпадащи с набор от зададени поведения, са аномалия. Аномалия е поведение, което не е нормално и е неправилно. За да разграничи аномални от нормални действия, IDS идентифицира мрежовата активност по профил и ако нещо не съвпада с профила, IDS го отбелязва като аномалия и генерира аларма.

Аномално-базирана IDS система работи чрез създаване на база от нормални действия в мрежата. Тази база е главно базирана на статистики, записани от поведението на входно-изходни операции, използване на процесора, паметта, потребителска активност и брой опити за влизане чрез идентифициране. IDS наблюдава мрежовата активност чрез сравняване на поведението на компонентите на мрежата с генерираната база. Ако е забелязано дори малко нарушение (от тези дефинирани в базата), аномално-базираната IDS генерира аларма.

Например използването на процесора на компютър в мрежата е било 70% през последния месец. Обаче, за един специфичен ден, нейният процесор е показал използване на 100%. За IDS това е необичайно поведение и следователно е генерирана аларма.

Една аномално-базирана IDS има обаче няколко недостатъка. Един от тях е, че всяко нарушение на нормалната активност се третира като пробив. Като резултат от това, дори и най-малкото нарушение генерира аларма и в повечето случаи тя е false-positive. Друг недостатък е, че този модел няма възможност да анализира поведение по шаблон и да идентифицира причината за ненормалното събитие.

Misuse-Detection техника

Misuse-detection техниката представя атаките във формата на шаблон или сигнатура. В този модел, IDS поддържа база от данни на всички от познатите сигнатури на атаки. Аларма се генерира винаги, когато сигнатурата на атаката съвпада с тази, която IDS има в нейната база данни. Misuse-detection техниката също има възможността да открива варианти на една и съща атака.

Моделът на злоупотреба действа като антивирусен софтуер, който открива всички вирусни атаки, ако е обновяван периодично с най-новите вирусни сигнатури: По подобен начин базираният на misuse-detection IDS модел открива само тези сигнатури, които вече са запазени в базата от данни.

В IDS, базирана на този модел, броят на false-positive алармите е нула. Обаче, затруднението в този модел е, че системата не може да открива нови типове атаки, които не са били открити по-рано.

Типове системи за откриване на пробиви

Системите за откриване на пробиви, базирани на модела на аномалия или злоупотреба, могат да бъдат класифицирани като:

- Мрежово-базирани IDS;
- Хост-базирани IDS;
- Хибридни IDS.

Тези класификации са базирани на начина, по който всяка една IDS е разгърната в мрежата. Всяка IDS може да бъде извън мрежата, като част от мрежата или и на двете нива.

Често използвани термини, свързани с IDS

Следните термини са компонентите на всяка IDS, независимо дали е мрежово-базирана, хост-базирана или хибридна.

Командна конзола

Командната конзола е централната среда, която контролира цялата IDS. Тя обикновено е посветена машина с инструменти за указване на политики и аларми. Командната конзола поддържа контакт с наблюдаваните машини и/или мрежовите сензори през кодирана връзка.

Командната конзола също така изпълнява функциите на оценяващ (assessment) мениджър, целеви (target) мениджър и сигнален (alert) мениджър. IDS може да има тези функции инсталирани на една конзола или разгърнати на отделни компоненти. В основни линии определящият мениджър се грижи за статичната конфигурационна информация, целевият мениджър поддържа връзките с компоненти по мрежата, а сигналният мениджър събира и поддържа сигналите за проблеми.

В повечето случаи, производителите осигуряват собствена конзола с техния продукт. Сега обаче има тенденция за интегриране на възможностите на командната конзола с контролните мрежови системи, като HP OpenView, и корпоративните контролни системи, като Tivoli.

Сензор

Сензорите са софтуерни програми, които могат да бъдат инсталирани на посветени машини или мрежови устройства в критични сегменти на мрежата. Това са самостоятелни системи за откриване, през които данните по мрежата минават. Главната роля на сензорите е да търсят в мрежовите пакети шаблони за злоупотреба и в случай на нарушение да генерират аларми в централната конзола на мрежата.

Оповестяване на проблем

Този компонент в IDS е отговорен за свързването с отговорника по сигурността в случай на забелязано нарушение. Това става посредством аларми на екрана, звукови оповестявания, изпращане на съобщения до пейджър или електронна поща.

Реагираща подсистема

Както предполага името, тя е отговорна за извършването на действие, базирано на отчетените заплахи към системата-цел. Тези подсистеми могат да генерират отговори автоматично или по заявки от системния оператор. Най-честите отговори (ответни реакции), генерирани от подсистемата, включват преконфигуриране на рутер или защитна стена и изключване на системата-цел.

База от данни

Базата от данни е информационен склад за всички дейности, наблюдавани от IDS. Базата от данни включва статистика и за злоупотребите, и за поведението. Тези статистики спомагат за създаването на поведенчески шаблон за действия, които по-късно могат да се използват за оценяване на вредата и изследване.

Мрежов кран

Мрежовият кран (network tap) е устройство или софтуерна програма, която позволява събирането на информация от мрежата. Когато е устройство кранът на мрежата трябва да има (поне) три порта: порт A, порт B и порт за монитор. Освен предаването на трафика между портовете A и B в реално време копира същите тези данни на порта на монитора, което позволява на трета страна да слуша.

Мрежово-базирана IDS

Мрежово-базираната IDS се използва за наблюдение и анализ на пакети от данни, които минават по мрежата. Този тип IDS действа в контраст с хост-базираната IDS, която оценява данни, произхождащи от компютрите (хостовете), като например журнали със събития. Мрежово-базираната IDS е позиционирана така, че да открива опити за достъп и различни атаки, произхождащи извън мрежата.

Повечето мрежово-базирани атаки целят злоупотреба чрез уязвимостите на операционната система, като неупълномощен достъп или влизане, кражба на данни или ресурси, изтегляне на пароли, кражба на трафик, наводнение с пакети, деформирани пакети, DoS и DDoS атаки (атаки с отказ на услуги). Тези типове атаки се откриват само от мрежово-базирана IDS и това е ключов фактор в различаването на двете технологии.

Архитектура на мрежово-базираната IDS

Мрежово-базираната IDS се състои от сензори, разгърнати из цялата мрежа и докладващи на командната конзола. Мрежово-базираната IDS може да е базирана на следните две архитектури:

- Традиционна сензорна архитектура;
- Разпределена мрежова архитектура.

Традиционната сензорно-базирана архитектура, също наричана мрежова IDS в безразборен режим (promiscuous mode), цели наблюдаването на всички сегменти от мрежата. В такъв случай сензорите в безразборен режим са разположени на специална машина. Разпределените мрежови архитектури наблюдават пакетите насочени към един компютър. За разлика от безразборния режим, разпределената архитектура има набор от агенти на важните машини.

Традиционна сензорна архитектура

Един сензор в безразборен режим се използва, за да надуши пакети, излизащи от мрежата. Веднъж излезли от мрежата, данните се подават на засичащ механизъм, който да прецени тяхната неприкосновеност. Надушващи механизми обикновено се инсталират на машината-сензор или на друг компонент от мрежата. Мрежови кранове се използват заедно със сензорите, за да вземат пакети от мрежата и ги разпределят към ключовите точки. Движението на данните през традиционната сензорна архитектура преминава през следните стъпки:

1. Когато два компютъра комуникират един с друг, се обменят пакети.
2. Пакетите след това се проверяват чрез сензор, който е сложен в мрежата, някъде между комуникиращите компютри.
3. Проверяващият механизъм се използва, за да сравни предефинираните шаблони с тези на текущия пакет. Ако данните съвпадат, се генерира аларма и се препраща към централната конзола.
4. Чрез конзолата отговорникът по сигурността е уведомен чрез различни методи - звуково, визуално, пейджър, електронна поща или SNMP.
5. Генерира се отговор от подсистемата автоматично или по заявка на отговорника по сигурността.
6. Алармата (с детайли за злоупотребата и шаблоните на поведение) се записва за по-късно разглеждане, корелация или оценка.

7. Генерира се отчет, обобщаващ активността.
8. Използват се дейта-детективи, за да преценят по-нататъшни тенденции. Някои IDS системи също така разрешават записване на оригинал трафик, така че сесията да може да се възпроизведе за преглед късно.

фигура 38 *Традиционна мрежова архитектура*

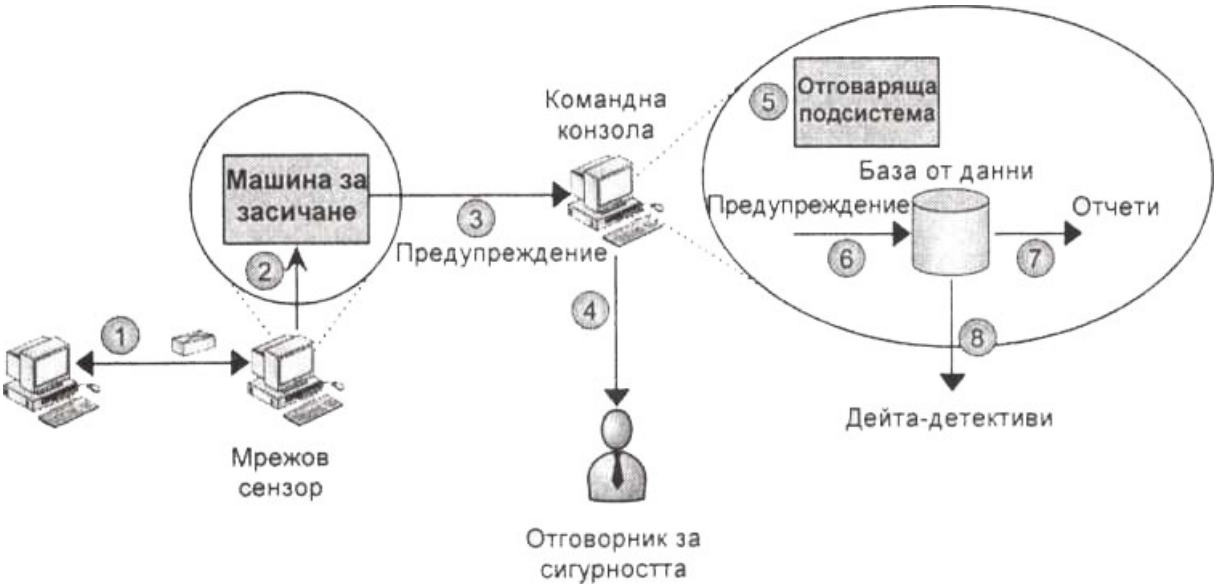
Разпределена мрежова архитектура

Мрежовата IDS система в безразборен режим предизвика загуба на пакети във високоскоростните мрежи. Като решение на това беше предложена разпределената мрежова архитектура на IDS. Тази архитектура има сензор, закачен на всеки компютър в мрежата. Всеки сензор се грижи само за пакетите касаещи компютъра, на който е инсталиран. След това сензорите комуникират помежду си и с главната конзола, за да сравнят и направят връзка между алармите.

Тази архитектура на мрежов възел доведе до объркване по отношение на разликите между мрежово-базирани и хост-базирани IDS системи. Ако мрежов сензор работи на хост, това не го прави хост-базиран сензор. Мрежовите пакети, преминаващи и надушвани при този хост, са все още част от мрежовата IDS. Основната разлика между мрежова и хост-базирана IDS не е позицията на сензора или режимът на работа, а произходът на данните. Мрежово-базираната IDS обработва TCP/IP пакети, докато хост-базираната IDS обработва журналите със събития, генерирани от операционната система и приложенията.

Движението на данните през разпределена мрежова архитектура преминава през следните стъпки:

1. Когато компютър се свързва с друг компютър, се обменят пакети.
2. Пакетите след това се проверяват чрез сензор, който е сложен на компютър, за който се проследява трафика.
3. Проверяващият механизъм се използва, за да сравни предефинираните шаблони с тези на текущия пакет. Ако данните съвпадат, се генерира аларма и се препраща към централната конзола.
4. Чрез конзолата отговорникът по сигурността е уведомен чрез различни методи.
5. Генерира се отговор от подсистемата автоматично или по заявка на отговорника по сигурността.
6. Алармата (с детайли за злоупотребата и шаблоните на поведение) се записва за по-късно разглеждане, корелация или оценка.
7. Генерира се отчет, обобщаващ активността.
8. Използват се дейта-детективи, за да преценят по-нататъшни тенденции.



фигура 39 *Разпределена мрежова архитектура*

Работен режим на мрежово-базираната IDS

Разполагането на сензор в мрежата играе важна роля в дефинирането на работния режим на IDS. Сензор, разположен извън защитна стена, разпознава адреса на източника, опитващ да се свърже с мрежата извън организацията. Сензор, разположен във вътрешната мрежа (зад защитна стена), открива атаките, които успешно прескачат вашата защитна стена.

Тези сензори са също полезни при откриване на забранен трафик, произхождащ вътре в мрежата, насочен към адрес извън вашата защитна стена.

Една мрежово-базирана IDS може да използва двата работни режима - с предупреждения (tip-off режим) и наблюдение (surveillance режим).

Tip-Off режим

В този режим IDS се използва, за да открие мрежов пробив по времето на случването му. Това е обичайният подход за откриване на пробив, при който шаблоните се наблюдават и всяка открита подозрителна активност се подава (като предупреждение) към отговорника по сигурността, показвайки, че е възможно да има пробив. Операционният tip-off режим основно е предупреждение, отбелязващо, че системата открива ненормална активност, която преди това не е откривана.

Surveillance режим

При surveillance режима, както подсказва името, се наблюдава машината за поведенчески шаблони за злоупотреба. Основното свойство на surveillance режима е наблюдението над поведението на малък набор от компоненти в мрежата. За разлика от tip-off режима, surveillance се предприема, когато вече е имало пробив, отбелязан е или е предположен.

Ползи от мрежово-базираните IDS

Мрежовите IDS са изчерпателен план срещу заплахи, произхождащи извън организацията. В комбинация с хост-базирани технологии могат да открият повечето заплахи и злоупотреби. Следват ползите от мрежова IDS.

Възпиране

Мрежовата IDS може да изпрати на хакерите бележка, че тяхната постъпка може да доведе до съдебен иск. Възпиращата стойност на IDS може да бъде подобрена чрез предприемане на мерки, като например електронна поща от системния администратор, настояваща прекратяване на атаката. Електронната поща може също да служи като предупреждение към други хакери, посочващо, че влизането в мрежата на организацията може да доведе до съдебен иск. Ако хакерът е достатъчно умен да промени адреса на подателя, възпирането може да не е ефективно.

Засичане

Мрежовата IDS засича дейности в контекста едновременно на определяне и на поддръжка при вземане на решение. Сигнатурите служат като определящи инструменти, защото откриват шаблони и ги сравняват с предефинирани такива. Експертите-оператори правят ефективен анализ на тези сигнатури, като използват инструменти за анализ на поведението. Статистическите методи служат като механизми за улесняване на решението. Тези методи може да не са в състояние да определят атакуващия и типа на атаката, но могат да доставят достатъчно информация, която при правилно използване може да спомогне за откриването на проблема.

Автоматична реакция и механизъм на оповестяване

Повечето IDS могат да реагират, когато открият пробив. Реакцията може да е автоматична или ръчна и може да включва едновременно локално и отдалечено известяване. Повечето търговски продукти осигуряват следните възможности:

- *Пейджър (Pager)*. Най-често използваното средство, защото известява администратора, независимо къде се намира той. Това е също рентабилен метод на известяване.
- *SNMP trap*. Известява центъра на операциите (сървър).
- *Визуален (Onscreen)*. Подава известяване на конзолата. Този метод изисква някой да наблюдава за бележки, примигващи на конзолата.
- *Звуков (Audible)*. Този метод е успешен, ако получателят е в обсега на звука на алармата.
- *Електронна поща (e-mail)*. Този метод е ефективен, само ако алармата не е критична и не изисква непосредствена намеса - може да почака ден или два.

Изброените средства могат да се използват, за да се генерират и автоматични, и ръчни реакции. Автоматичните реакции имат присъщи рискове и поради тази причина трябва да се използват внимателно. Повечето търговски продукти осигуряват следните възможности за автоматична реакция:

- *Преконфигуриране на рутер/защитна стена*. Преконфигурирането на рутер или защитна стена включва

отказване или блокиране на адресите, опитващи атаки срещу мрежата.

- *Ответна атака*. Някои IDS са конфигурирани да атакуват в случай на открит пробив. Такава възможност обаче не се препоръчва заради законовите постановки.
- *Затваряне на връзката*. Друга автоматична реакция е да се затвори връзката на атакуващия, докато той още не е извършил атаката.

Проблеми в мрежово-базираната IDS

Както казахме по-рано, традиционните мрежово-базирани технологии срещат проблеми като загуба на пакети във високоскоростни мрежи, switched мрежи и криптиране. В допълнение, навлязоха нови технологии, които могат да открият следяща програма (sniffer), разположена от IDS за прихващане на данни. Това превърна прихващащите устройства в цели за атаки.

Реасемблиране на пакети

Преди изпращане на съобщение, TCP/IP разделя съобщението на малки пакети, които след това трябва да се реасемблират при компютъра-получател. Няколко мрежово-базирани сигнатури правят търсене в съдържанието на пълното съобщение. Тъй като съобщението е разчупено на малки парчета, пакетите може да заобиколят машината за засичане и съобщението да се окаже опасно при асемблирането му при получателя. Решението на този проблем е да се използва хост- базираната архитектура. Тъй като всеки сензор е разположен на системата- получател, пакетите се реасемблират на системата и сигнатурата се сравнява с цялото съобщение.

Високоскоростни мрежи

Оказа се, че мрежовите кранове (taps), използвани в стандартните технологии, изпускат пакети при високоскоростни мрежи. Ако мрежата е натоварена с трафик, проблемът се влошава. Например, кран на 10 MB TCP/IP мрежа не показва никакъв проблем да поддържа скоростта на придвижване на данните. Но кран, разположен на 100 MB мрежа, която е силно натоварена, може да изпусне много пакети, които от друга страна са необходими за засичане. Мрежи, по-бързи от 100 MB, създават повече проблеми.

Пакети, изпуснати от мрежата, могат да осигурят прикритие за хакери. Хакерите умишлено наводняват мрежата, за да накарат системата да изпуска пакети. Обаче мрежа, наводнена с пакети, е признак за откриваемо поведение.

Архитектурата на мрежов възел е ефективно решение на този проблем. Сензорите, разположени на машината-цел, лесно могат да се справят с пакети от данни.

Sniffer-detection програми

Съществуват програми, способни да открият следяща програма (sniffer), разположена в мрежата. Има например програма, наречена AntiSniff, която изпраща пакети с различни форми на всеки компютър от мрежата. Тези пакети използват латентността и други техники, за да преценят дали машината има инсталирана следяща програма, например в случай на мрежово-базирана IDS, за да открият дали има мрежов кран.

Програми като AntiSniff обещават да открият мрежови кранове, използвани от IDS. Когато бъде открит мрежов кран, атакуващият има достатъчно информация, за да избегне системата за засичане.

Криптиране

Повечето IDS зависят от сравняването на сигнатури за откриване на пробиви. Ако изпращаните пакети с данни са криптирани, тогава съвпадението на шаблон е невъзможно. Криптирането обаче става обичайно в повечето нива на мрежите. Например то се ползва във виртуалните частни мрежи (VPN), secure shell (ssh) и secure socket layer (SSL). Възможни са решения за надхитряване на криптирането. Обаче никое от тях не е пълно решение на тези ограничения. Следват няколко решения за заобикаляне на криптирането, докато откривате пробиви:

- Разполагане на мрежовите сензори вътре във VPN устройството, където данните се декриптират. Това решение обаче работи само донякъде, защото през по-голямата част от времето данните са криптирани на ниво сесия или приложение. Това показва, че няма място, останало за IDS да вземе данните в декриптиран вид от мрежата.
- Записване на криптиращи ключове на маршрутизатор или други мрежови устройства. Това също не работи много добре.

Управлението на криптиращи ключове за декриптиране на данни не е много лесна работа и е голяма заплаха за сигурността.

Хост-базирани IDS

Една хост-базирана IDS система използва данни за анализ, произхождащи от компютри (хостове), като журнали на приложения и системни събития. Тази система не е като мрежово-базираната IDS, която използва данни, произхождащи от мрежата, като TCP/IP пакети.

Източниците на данни в хост включват журналите на събития в операционната система (като ядро, основен модел на сигурност и сигурност) и журналите на приложения (като системен журнал, релационни бази от данни и Web сървър). Хост-базираните технологии са ефективни в откриването на злоупотреба в мрежата, защото данните, които се използват за анализ, са разположени на машината с упълномощени потребители. Журналите на събития от тези машини осигуряват информация относно достъпа до файлове и използването на програми от упълномощени потребители. Тази близост до упълномощените потребители дава на администраторите възможност да анализират тенденции и да предприемат оценка на вредата върху данните. Тъй като журналите на събития са върху доверен източник и са защитени, те могат да бъдат представени в съда като поддържаща документация, докато се преследват компютърни престъпници.

Цената на разгръщането на хост-базирана IDS е по-висока от традиционната мрежова IDS. Обаче цените се минимизират, ако хост-базираното наблюдение се поддържа правилно.

Атаки, откривани чрез хост-базирана IDS

Примери:

- *Злоупотреба с привилегировани права*. Това се случва, когато на потребител са дадени root права, административни или други привилегии и той ги ползва за незаконни цели. Хост-базираната IDS е успешна в такива ситуации, защото наблюдението се извършва на същата система, на която правата са били дадени.
- *Злоупотреба с повишени привилегии*. Администраторите обикновено дават повишени привилегии на потребители с цел

инсталация на специфични приложения, ускоряване на работния процес или достъп специфичен файл от мрежата или специфична машина. Повечето политики на сигурността ограничават root или администраторски привилегии, но често възникват ситуации, когато администраторите трябва дадат тези разрешения. Така администраторът повишава правата на потребители, мислейки, че ще ги намали по-късно. Много пъти обаче администраторът забравя да премахне тези привилегии, което може да доведе до злоупотреба с тях. В допълнение, една заявка за повишаване на права може да съдържа намерения за социален инженеринг, целяща достъп до други ресурси на системата и мрежата.

- *Използване на акаунт от бивши служители*. В момента, в който един служител не е част от организацията, повечето организации има политика на изтриване или отнемане на акаунт. Обаче изтриването или отнемането на акаунт може да отнеме известно време, а това може да остави вратите отворени за служители, които могат да експлоатират ситуацията в тяхна полза.
- *Създаване на акаунти със задни вратички*. Може да се създадат ситуации, при които администраторът създава акаунт, за който само той знае. Например, докато инсталира софтуерен пакет, за успешно изпълнение, софтуерът може да изиска създаването на акаунт. В нормалния случай, за създаването на акаунт се получава формална заявка и след се попълва подходящата документация. Но по време на инсталацията администраторът може да създаде акаунт, без да следва формалната процедура. Може да дойде време, когато от администратора е поискано да напусне организацията и така привилегиите, свързани с неговите акаунти, са отнети. Но тъй като няма документация за един от акаунтите, администраторът има права над акаунт, относно който организацията не е наясно. Това оставя отворена дупка.

Архитектура на хост-базираните IDS

Има две възможни архитектури за хост-базирана IDS: *централизирана* и *разпределена* в реално време. Тези архитектури са базирани на разпределени целеви агенти (target agent).

Целеви агент

Целевият агент е малка програма, която върви на системата. Целевият агент разрешава на системата да изпълнява привилегировани дейности локално, които иначе не биха били възможни. Някои от тези дейности включват прихващане на TCP/IP пакети от мрежата, обработване на данни от журнали на събития, централизиране на необработени данни от журнали, проверка на целостта на файловете, проверка на системната конфигурация, откриване на злоупотреби, препращане на аларми (предупреждения) и изпълняване на реакции локално в целевата машина.

Агентите обикновено работят във фонов режим в UNIX или като услуги в Windows хостове. Тези хостове имат комуникационни програми, като програмни интерфейси на приложенията (APIs) за отдалечена администрация, които осигуряват възможностите на агентите. Тези хостове сами могат да извършват функциите на агенти (въпреки че не много ефективно, освен ако малките програми-агенти не са инсталирани). Въпреки това, веднъж след като хост има инсталирана изпълнима програма-агент, неговата способност да работи като упълномощен локален потребител не се намалява. Правата на крайния потребител върху хост с инсталиран агент не се променят, ако агентът е конфигуриран и се поддържа правилно.

Възможно е да има и един, и няколко агента на една система. Единичен агент може да осигурява множество възможности, а в същото време множество агенти могат да изпълняват една-единствена задача върху хоста.

Централизирана хост-базирана архитектура

В централизираната хост-базирана архитектура, необработеният журнал на събитията се изпраща на централно място (на машина, различна от хост), преди да бъде анализиран. Функционирането на архитектурата преминава през следните стъпки:

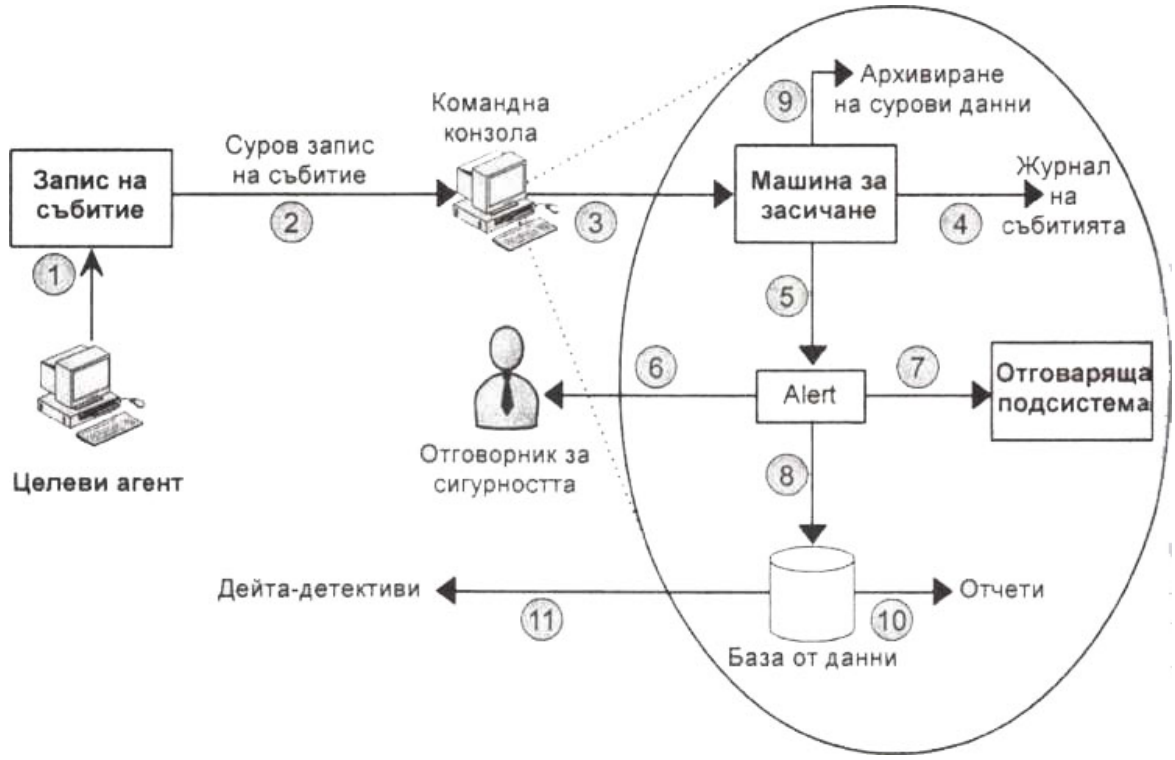
1. Когато в системата се предприеме действие, като достъп до файл или изпълнение на програма, като например MS Word, се създава запис на събитието. Записът се добавя във файл, който обикновено е затворен в доверената база от данни на системата.
2. Целевият агент изпраща файла на централизираната командна конзола в предварително указани времеви интервали през сигурен комуникационен канал.

3. Откриващият механизъм (върху конзолата) сравнява шаблона на поведението на този файл с предефинираните шаблони. Записите на данните са сравнени в суров формат, който всъщност е техният оригинален формат.
4. Едновременно с това се създава журнал на събитието. Този журнал се използва като архив на данни за организацията в случай на съдебно преследване.
5. Ако шаблонът на файла съвпадне с предефинираните шаблони, се генерира аларма и се препраща на различните подсистеми за оповестяване, реакция и запазване.
6. Отговорникът по сигурността се уведомява чрез звукови или визуални методи като пейджър, електронна поща, SNMP trap или друго.
7. Генерира се реакция. Подсистемата за реакции сравнява алармата с предефинираните реакции или приема заявка от отговорника по сигурността.
8. Алармата се запазва в релационната база от данни. Базата от данни може да пази статистическа информация в допълнение към алармите.
9. Суровите (необработените) данни се прехвърлят към архив за такива данни. Периодично се прави резервно копие (backup) на такъв архив, за да се намали количеството използвано дисково пространство.
10. Генерират се отчети, обобщаващи алармите и дневниците със събития. Отчетът също така има детайли за включената целева система и метода на атака.
11. Дейта-детективи се използват за оценяване на дългосрочните тенденции. Поведението се анализира чрез използване и на запазените данни в базата от данни, и на архива на суровия журнал на събитията.

Тази архитектура създава малко или никакво натоварване на системата, тъй като целият анализ се извършва на централната конзола. Няма ефект върху производителността на системата. В допълнение, това разрешава засичане в големи размери, защото има по-малко проблеми за производителността. Сравняването на няколко хост сигнатури едновременно е възможно, тъй като засичащият механизъм стои на централния хост и има достъп до данните на всички целеви системи. Накрая, централизираните сурови данни могат да се използват за съдебни цели и дейта-детективите могат да оценят дългосрочните тенденции.

Недостатъците на тази система включват липсата на засичания и генериране на реакции в реално време. Генерира се мрежов трафик, докато

данните се централизират на конзолата. Ако ширината на канала на мрежата вече е запълнена до краен предел, мрежата може да се задръсти след време.



фигура 40 *Централизирана хост-базирана архитектура*

В таблица 4 са обобщени предимствата и недостатъците на централизираната архитектура.

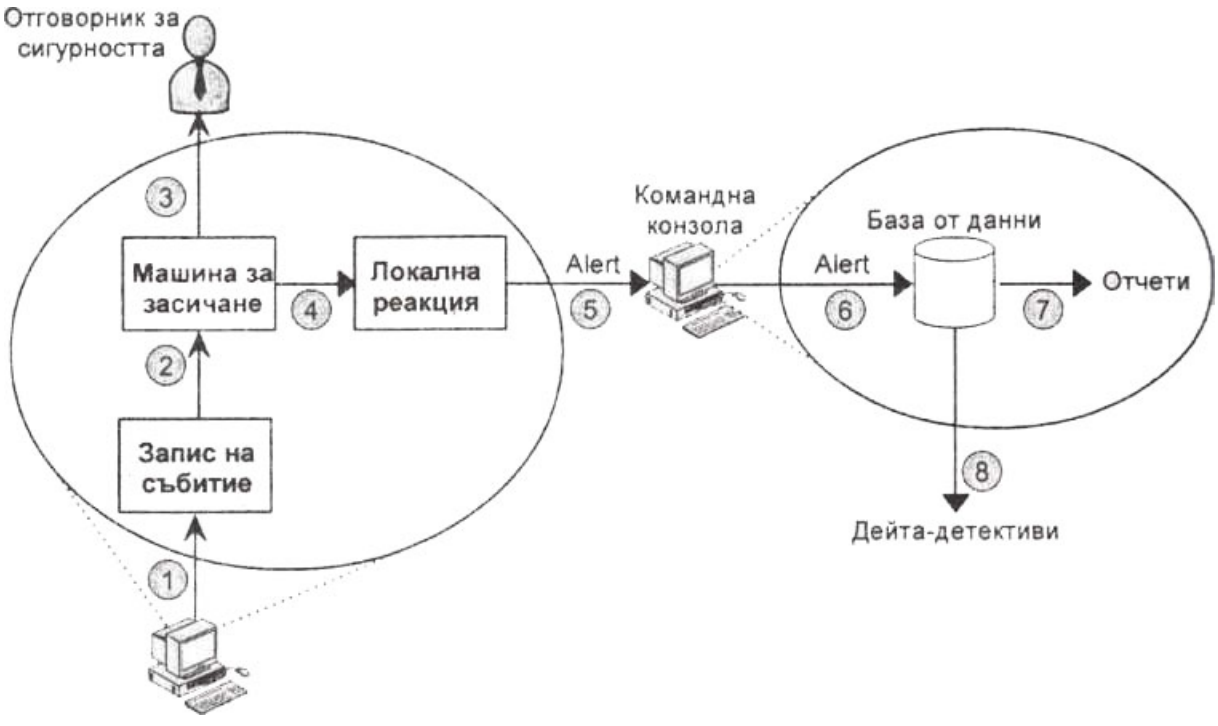
Предимства	**Недостатъци**
Няма ефект върху производителността на системата	Няма засичане в реално време
Предоставя статистика на поведението	Няма реакция в реално време
Възможно е сравняване на няколко сигнатури едновременно	Генерира товар по мрежата
Предоставя сурови данни от архивите за съдебно преследване	

таблица 4 *Предимства и недостатъци на архитектурата за централизирано засичане*

Разпределена архитектура в реално време

В разпределената архитектура, суровият журнал на събитията се анализира на машината на целевия агент. Функционирането на архитектурата преминава през следните стъпки:

1. Когато се предприеме действие в системата, се генерира журнал на събитията.
2. Журналът на събитията се записва във файл в реално време и се обработва само на системата, тъй като засичащият механизъм е на нея.
3. Уведомява се отговорникът по сигурността чрез звукови или визуални методи като пейджър, електронна поща, SNMP trap или различни други методи.
4. Генерира се реакция.
5. Генерира се аларма и се изпраща на централната конзола.
6. Алармата се запазва в релационна база данни. Статистическите данни за поведението не са достъпни при тази архитектура, тъй като данните са ограничени само до една система.
7. Генерират се отчети, обобщаващи алармите и дневниците със събитията. Отчетът също включва детайли на включените целеви системи и метода на атака.
8. Използват се дейта-детективи, за да оценят дългосрочните тенденции Поведението се анализира чрез използване само на данните, записани в базата от данни, защото няма генериран архив със сурови данни.



фигура 41 *Разпределената архитектура в реално време*

Основното предимство на тази архитектура е реално-времевата обработка на данните. Основният недостатък е, че производителността на машината-хост да бъде афектирана ако агента не е добре управляван и

разгърнат. В таблица 5 се обобщават предимствата и недостатъците на разпределената архитектура в реално време.

Предимства	Недостатъци
Аларми в реално време	Може да има отражение върху производителността.
Реакции в реално време	Няма статистика на поведението.
	Натоварва мрежата
	Не е възможно сравняване на няколко сигнатури.
	Няма поддръжка на архиви със сурови данни

таблица 5 *Предимства и недостатъци на архитектурата в реално време*

Работни режими на хост-базираните IDS

Хост-базираната IDS може да се използва в четири работни режима: с предупреждения (tip-off), наблюдение, оценка на щетите и потвърждение. Следващият раздел обсъжда два от тези режими: оценка на щетите и потвърждение. Първите два режима са разгледани при мрежово-базираните IDS.

Оценка на щетите

Оценката на щетите, както предполага името, включва идентифициране на степента на вредата, причинена от злоупотребата. Тъй като се поддържа база със записите на събитията, лесно могат да се изведат действията, довели до злоупотреба, нейната област, вредата над сигурността и произхождащите от това ефекти. Възможностите на системата, като докладване, архивиране на сурови данни и дейта-детективи, спомагат в оценяването на щетите.

Този работен режим е особено полезен, ако човешките ресурси са недостатъчни. Следователно хост-базираните системи могат изключително да се използват за оценка на щетите, ако желаете да намалите извънредна работа.

Потвърждение

Този режим потвърждава, че потребителите следват политиките по сигурността. Поведенческото наблюдение се използва като допълнителен инструмент, за да се следят действията на потребителите. Режимът “потвърждение” извежда определени проблеми на повърхността, които

иначе може да останат незабелязани. Пример за такъв проблем е потребител се идентифицира през нощта, опитвайки се да прочете забранени файлове, или неправилното използване на приложения или процеси.

Ползи от хост-базираната IDS

Ползите от хост-базираната IDS включват откриване на злоупотребата, възпиране, очакване на злоупотреба, реагиращ механизъм и оценка на щетите. Тези ползи бяха разгледани в предишните раздели, които са почти същите като при мрежово-базираните IDS. Има обаче една уникална полза, която не присъства в мрежово-базираната IDS и която предпазва от злоупотреби от вътрешната мрежа, обикновено позната като възпиране на вътрешни хора (insider deterrence).

Тази полза е подобна на сигурността, осигурена от видеокамерата в бижутерски магазин. Ефектът на възпирането е подобен на принципа, че хората са по-бдителни, ако знаят, че са наблюдавани. Insider deterrence е всъщност едно от най- големите предимства на хост-базираната IDS, защото тя избягва злоупотреби на ресурси, тъй като служителите са наясно, че ги наблюдават през цялото време.

Организацията може да увеличи ефекта от възпирането по няколко начина. Един от тях е да оповести съществуването му. Както хората слагат табели за охранителни системи на вратите си, по същия начин организациите могат да сложат банери за упълномощаване на служителските компютри, които банери да напомнят на служителите, че действията им се наблюдават продължително. Този метод също би елиминирал наблюдението на всеки компютър по всяко време, тъй като служителите вече са внимателни. Друг метод за увеличаване на ефекта от възпирането е да се покаже на служителите, че се наблюдават. В случай на злоупотреба, е необходимо служителят да бъде предупреден, за да е наясно, че някой го наблюдава.

Друг начин за увеличаване на ефекта е съдебният. Хост-базираните IDS осигуряват доказателство за злоупотреба чрез дневниците на събитията и статистически данни. Това доказателство по принцип е достатъчно за предприемане на законни мерки срещу нарушителя в допълнение към изпитателния срок и/или изгонването от организацията.

Продължителното наблюдение е очевидно приложима мярка за сигурност. В някои организации, като важни правителствени учреждения, такава сигурност би била гарантирана. Обаче този тип продължително наблюдение в средностатистическа организация може да има отрицателно влияние върху служителите (за разлика от други мерки, които работят във фонов режим). Администраторите, които биха използвали тези методи, трябва да са наясно, че такива преки и крайни методи на обвинение често се отразяват на служителите (вероятно и на производителността им).

Проблеми на хост-базираната IDS

Ползите, осигурени от хост-базираната IDS, също са придружени от някои разходи. Повечето от тези разходи могат да бъдат намалени от добра система, добре спазвана политика и добре планирани настройки на работното място. Обаче, от гледна точка на хост-базираните IDS, трябва да се предвидят проблемите с производителността, разгръщането и други неща с възможни проблеми.

Производителност

Както вече сте наясно, хост-базираните IDS са разпределени механизми, които обработват данни от машината, на която са разположени. Поради тази архитектура на хост-базирано засичане, производителността на този хост може да спадне. Намаляването на производителността не може да се избегне, но може да се контролира чрез правилно разгръщане, избор на архитектура и политика.

Разгръщане и поддръжка

Разгръщането на хост-базирана IDS е трудно, защото хостовете са широко разпределени (на разстояние). Всяка машина трябва да бъде с инсталирани агенти. Началната инсталация и поддръжката са времепоглъщаща работа и следователно изисква разпределяне и механизми за обновяване от разстояние.

Компрометиране

Самата цел на инсталирането на IDS може да бъде обезсмислена, ако хакер получи достъп до наблюдаваната система и изключи агента. Това би означавало компрометиране на откриващата (засичащата) система.

IDS системите не са много ефективни при откриване на първата злоупотреба или пробив, но са много ефективни срещу поведение, което вече е било дефинирано. Една добра система трябва да може да открие, че агентът е бил изключен. Журнал, отбелязващ систематичен шаблон от изключвания, може да означава, че набор от системи са под атака.

Манипулиране на записите от агенти

Хакери могат да получат достъп до агентите, за да променят записите в системата. Те може да вмъкнат записи в звуковия поток, който да наподобява телефонна активност, или да махнат записи, за да прикрият неупълномощената дейност. За целта могат да се използват доверени и защитени дневници на събитията, например двоичен журнал на ядрото (binary kernel log).

Лекция 6

Сигурна автентикация

По време на влизането си в системи, локално или през мрежа, ние удостоверяваме самоличността си по няколко пъти на ден. Например, докато комуникираме чрез e-mail, пазаруваме онлайн или посещаваме частен Web сайт, ние предоставяме някакво доказателство за това за кои се представяме. Това доказателство служи само за потвърждение на нашата самоличност пред системата Удостоверяването на автентичността (автентикацията) е процедура, проверяваща дали потребителят отговаря на самоличността, с която се представя. С други думи, това е механизъм, който потвърждава самоличността на потребителя или процеса, опитващ се да се свърже или влезе в системата.

Разграничаването на самоличността на процес или потребител от злонамерен процес или действащ нарушител е трудна задача. Тя включва сложни механизми, базирани на криптография.

Удостоверяването на автентичността се занимава с важния въпрос дали потребителят е този, за когото се представя, или е измамник. Чрез процеса на автентикация се осигурява свързване към компютъра единствено на валиден потребител с предварително зададени права за достъп. Удостоверяването на автентичността е първата стъпка към разрешената употреба на ресурси.

Упълномощаването проверява дали опитът на потребителя да използва определено приложение или файл е разрешено или не. Тоест, упълномощаването е стъпката, следваща автентикацията. След като потребителят е бил удостоверен и се опита да използва приложение или файл, получаването на разрешение се базира на правата за достъп, зададени за това приложение или файл, за конкретния потребител. Упълномощаването или контролът на достъпа също е много важен процес. Но поради факта, че процедурата е специфична за различните операционни системи, не може да бъде подробно разгледана, както механизмите за удостоверяване. Упълномощаването основно покрива концепции, свързани с проверяване, права за достъп до файлове и директории, правата на потребителя, както и квотите на диска.

Криптография

Криптографията е механизъм, който защитава информацията чрез криптиране (шифриране) и декриптиране (дешифриране) на съобщения с таен код или шифър. В историята криптографията е била използвана за осъществяването на сигурна, лична комуникация между хора, военните сили, правителствени агенции и дипломатически групи. Днес обаче, криптографията е основата на модерните технологии за сигурност, които защитават информацията и ресурсите в мрежата.

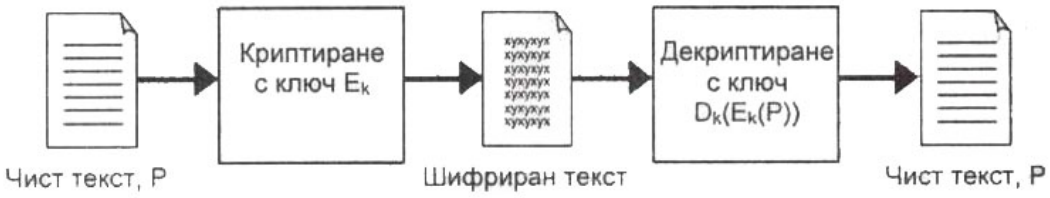
Една примерна система за свързване на чист текст, шифриран текст и закодиращ ключ може да се изрази по следния начин:

$$C = E_к(P)$$

Тази система предполага, че закодирането на чистия текст P, използвайки ключ к, дава шифрирания текст C. Аналогична е и следната система, която показва декодирането на съобщение:

$$D_k(E_k(P)) = P$$

Тази система предполага, че E и D са математически функции, където $D_k(E_k(P))$ декодира шифрирания текст $E_k(P)$ в чистия текст P (фигура 42).



фигура 42 *Криптографска последователност*

За разлика от ранната криптография, днешната е нещо повече от тайно писане. Тя прекрачва отвъд елементарните криптографски функции, като криптиране и декриптиране, и разполага с механизми за свързване на документи с датата на създаването им, за използване на споделени дискови устройства и за създаване на подсигуряващи механизми от високо ниво. Криптографията също така се е превърнала в неотделима част от процеса на автентикация.

В допълнение, криптографията може да се използва и за други цели. Криптографските инструменти могат да се използват за създаването на протоколи и механизми, които предоставят потребителски услуги, като например плащане по електронен път, без то да бъде излагано на риск.

Модерната криптография е основно насочена към разрешаването на трудни проблеми. Един проблем може да е труден, защото решението му изисква секретна информация като закодиране и декодиране на съобщение или подписване на цифров документ. Проблемът може да е труден и защото изисква изчисляването на сложни стойности, като например хеш стойностите за съобщение.

Цел на криптографията

Криптографията като цяло е свързана със сигурността и поверителността на информацията. Въпреки това, целта на криптографията е отвъд поверителността. Следват четирите основни функции, които криптографията се стреми да предоставя:

- *Поверителност*. Осигурява се възможност единствено за упълномощените потребители да могат да видят или използват секретна информация. Съществуват няколко инструмента, които могат да позволят на нарушители да подслушват мрежовия трафик и да прехващат ценна, лична информация. Криптографията предоставя механизми и техники, които осигуряват поверителността на информацията по мрежите.
- *Автентикация*. Осигурява проверяването на самоличността на обектите, комуникиращи през мрежата. Криптографията не само удостоверява самоличността на подателя на съобщението, но също така проверява и тази на получателя му.
- *Цялостност*. Осигурява проверката на оригиналното съдържание на информацията за манипулации или промени по време на предаването. Ако целостта на информацията не се проверява, някои може да я промени, при това незабелязано. Криптографските системи използват техники и механизми, които помагат при проверяване целостта на информацията, обменяна по мрежата. Пример за пробив на целостта на информацията би бил нарушител, фалшифициращ цифровия подпис на даден документ.

- *Признаване*. Прави невъзможно за една комуникираща група да отхвърли част от комуникация или цялата комуникация, приключила преди време. Ако признаването на транзакциите през мрежата не се следи, всеки би могъл да комуникира и впоследствие да отхвърли наличието на извършената комуникация. Пример за анулиране на признаване би бил подател на информация, непризнаващ се за неин автор. Подобен пример би бил получател на информация, отричащ факта, че я е получил.

Видове криптиране

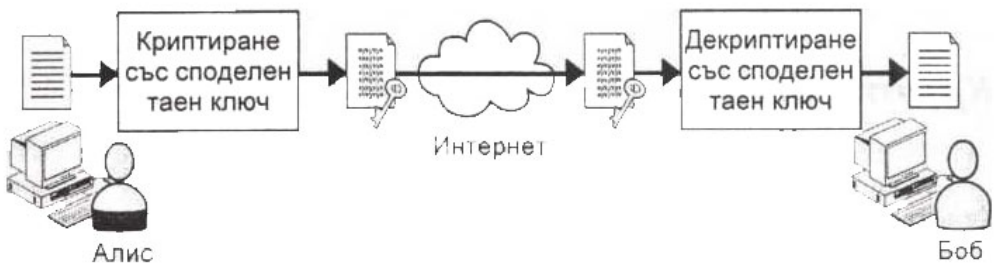
Двата основни вида алгоритми за криптиране са:

- Криптиране със симетричен ключ;
- Криптиране с несиметричен ключ.

Криптирането със симетричен ключ използва един и същ ключ за криптиране н декриптиране на съобщението. Този тип механизъм на криптиране се нарича механизъм за криптиране с таен ключ, защото ключът се споделя между подателя и получателя на съобщението. Споделянето на ключа се извършва по сигурен и таен начин, за да се поддържа поверителността.

Следният пример, обяснява механизма на криптиране със симетричен ключ. Алис иска да изпрати съобщение на Боб. Тя изпраща криптиране съобщение чрез извършване на следните стъпки (фигура 43):

1. Първо, Алис и Боб споделят копие на тайния ключ.
2. След това Алис криптира съобщението си с ключа и изпраща криптираното съобщение на Боб.
3. Накрая Боб декриптира съобщението със споделения ключ.



фигура 43 *Криптиране със симетричен ключ*

Честа употреба на криптирането със симетричен ключ има в протоколите на сигурността. Протоколите на сигурността използват симетрични ключове като сесийни ключове за поддържане на поверителността на съобщения в онлайн комуникация. Например, протоколите Transport Layer Security (TLS) и Internet Protocol Security (IPSec) използват механизма на симетричния ключ, за да генерират сесийни ключове със стандартни алгоритми за криптиране и декриптиране на съобщения. Всяка сесия поддържа различен сесиен ключ и тези ключове се подновяват на определени интервали.

Друга употреба на криптирането със симетричен ключ е в технологиите, които предоставят криптиране на голямо количество данни като например e-mail съобщения и документни файлове. Примери за тези технологии са Secure/Multipurpose Internet Mail Extensions (S/MIME), използваща симетрични ключове за криптиране на съобщения за поверителна електронна поща, и Encrypting File System (EFS), използваща симетрични ключове за криптиране на файлове за поверителност.

Сравнено с криптирането с несиметричен ключ, криптирането със симетричен ключ е от 100 до 1000 пъти по-бързо и натоварва процесорите по-малко. Причината се крие в това че, при криптирането с несиметричен ключ се използват по-тежки алгоритми в сравнение с криптирането със симетричен ключ. Поради тези причини криптирането със симетричен ключ се използва, за да предостави тайна там, където се изисква масивно криптиране и декриптиране.

Няколко недостатъка на криптирането със симетричен ключ:

- Ключът трябва да се споделя между страните, които искат да взаимодействат. Това може да отслаби сигурността, тъй като повече от един човек знае за ключа.
- Процесът на споделяне на ключа включва риска от пресрещане на ключа по мрежата от трето лице.
- Рискът се повтаря всеки път, когато ключът се променя или между страните.

Криптиране с несиметричен ключ

За разлика от криптирането със симетричен ключ, криптирането с несиметричен ключ използва различни ключове за криптиране и

декриптиране на съобщението. Криптирането с несиметричен ключ е по-известно като криптиране с публичен ключ.

Криптирането с публичен ключ изисква два ключа: публичен ключ и частен ключ. Частният ключ е известен само на собственика. Публичният ключ се споделя и е достъпен за всички страни, които биха искали да взаимодействат със собственика на частния ключ. Криптирането с публичен ключ действа чрез публикуване на публичния ключ на потребителя в директория (обикновено LD или X.500-съвместима директория). Тази директория е достъпна за всички потребители, които са заинтересувани да обменят съобщения.

Алгоритмите, използвани в криптирането с публичен ключ, включват сложни математически функции, които подсигуряват, че веднъж изпълнен, процесът криптиране не може да се обърне лесно. Тези алгоритми са проектирани по такъв начин, че съобщението, криптирано с публичен ключ, може да се декриптира само със съответния му частен ключ. По подобен начин съобщението, което е криптирано с частен ключ, може да се декриптира само със съответния публичен ключ от двойката. Освен това ключовете са генерирани по такъв начин, че не е възможно да определите единия ключ, дори и да знаете другия.

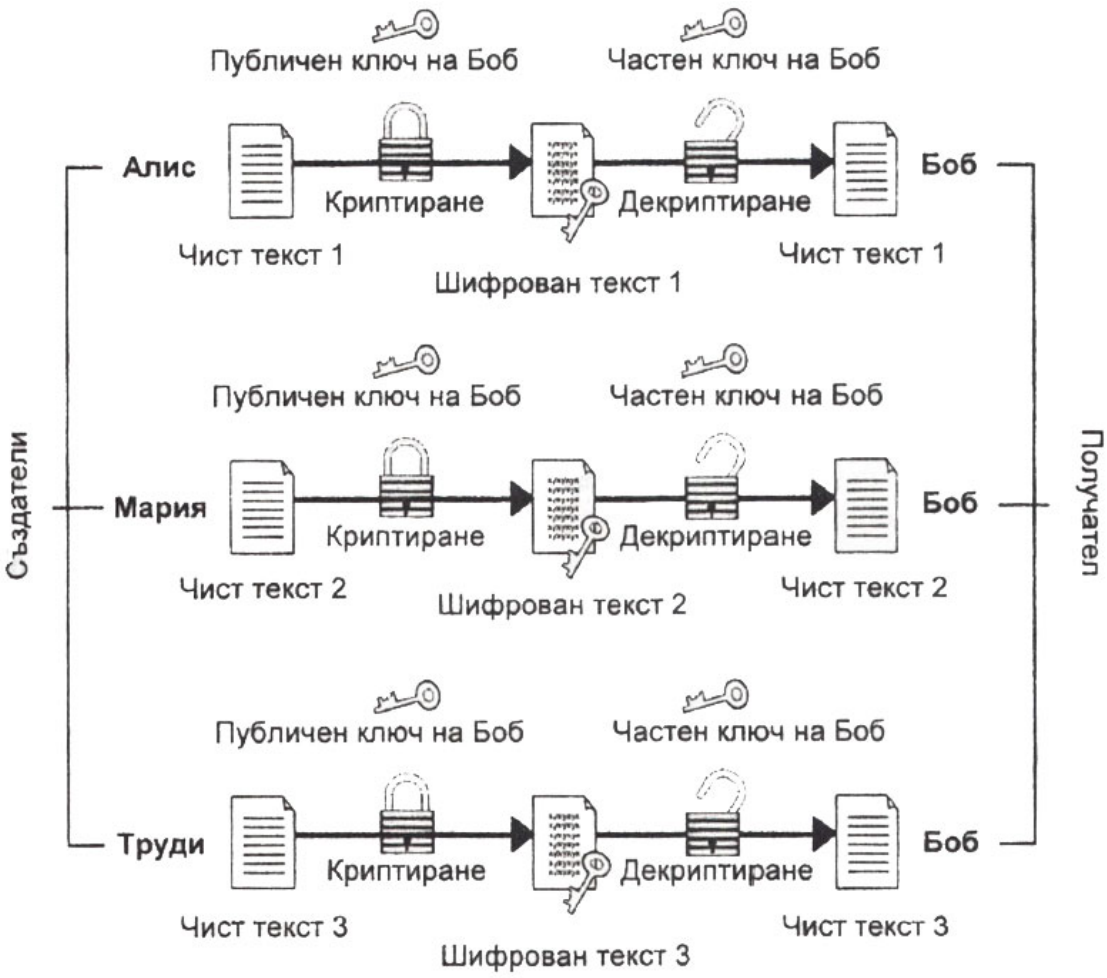
Основен принцип на несиметричното криптиране: от гледна точка на изчисленията е невъзможно ключът за декриптиране да бъде определен само с познаването на криптографския алгоритъм и ключа за криптиране.

Следният пример, представя механизма на криптиране с несиметричен ключ (фигура 44). Алис иска да изпрати съобщение на Боб. Тя изпраща криптиране съобщение, като преминава през следните стъпки:

1. Алис получава копие от публичния ключ на Боб, който е публикуван в директория в мрежата от Боб.
2. Алис криптира съобщението с публичния ключ на Боб и му го изпраща.
3. Накрая, Боб декриптира съобщението с частния си ключ.

Често срещана реализация на този механизъм е RSA криптирането с публичен ключ. Следват функции, които RSA криптирането с публичен ключ може да изпълни:

- Криптиране на симетричните тайни ключове, за да ги защити по време на обмена им по мрежата или докато се използват, съхраняват или хешират в системи.
- Създаване на цифрови подписи, които предоставят автентикация и признаване за ресурси и цялостност на данни за електронни документи и съобщения.



фигура 44 *Криптиране с несиметричен ключ*

Алгоритми за криптиране

Модерните алгоритми за криптиране се базират на един и същ основен дизайн (субституция и транспозиция - замяна и разместване) като традиционните алгоритми, но все пак алгоритмите не са еднакви. Традиционните алгоритми бяха прости, но използваха дълги ключове, за да разработят механизма на крипти Модерните алгоритми работят по напълно противоположна теория. Алгоритмите, които се използват в наши дни, са направени толкова сложни, че дори и специалист по криптиране да успее да

получи криптиран текст, ще му бъде трудно да го декриптира или да извлече някакъв смисъл от него поради сложността му. Въпреки това нито един код не е неразбиваем или неразгадаем.

Пример за симетричен алгоритъм може да бъде Data Encryption Standard (DES) създаден от IBM и приет за използване през 1977 г. През 2001 г. е заменен от Advanced Encryption Standard (AES). За учебни цели би могъл да се разгледа неговият опростен вариант Simplified Data Encryption Standard (S-DES). Отделно международен алгоритъм за криптиране на данни IDEA (International Data Encryption Algorithm).

Пример за криптиране с публичен ключ може да бъдат алгоритмите Diffie-Hellman и RSA.

Механизми за автентикация

Има няколко достъпни механизма, които помагат за проверката на самоличността на потребител по време на идентифицирането му в системата. Следват няколко протокола, които предоставят услуги за проверка на автентичност:

- Challenge-response протоколи;
- KDC механизъм за автентикация;
- Kerberos;
- Цифрови подписи.

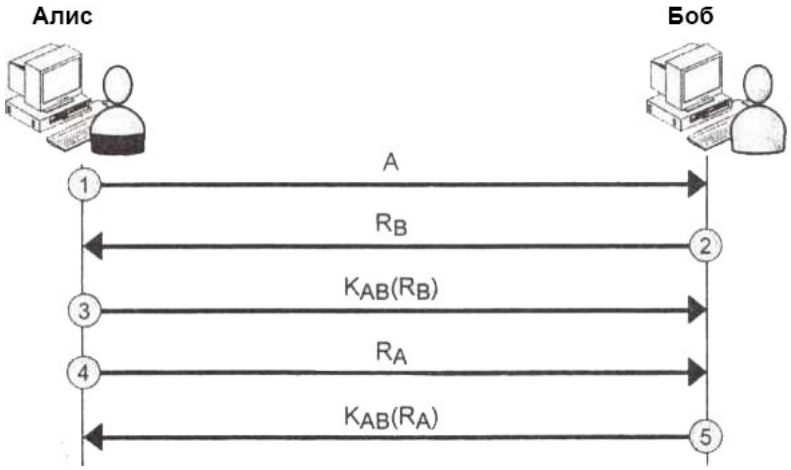
Challenge-Response протоколи

Challenge-response протоколите се базират на принципа, че едната страна изпраща произволно число като предизвикателство (challenge) до другата, която го обръща в специално число и връща резултата. За по-добро обяснение на challenge-response протокола отново ще се разгледа пример с Алис и Боб (фигура 45). Приема се, че тайният ключ вече е споделен помежду им.

1. Алис изпраща самоличността си А на Боб в съобщение 1. Съобщението се изпраща по начина, по който са се разбрали Алис и Боб.
2. На този етап Боб не е сигурен за самоличността на Алис. За да се подсигури, че съобщението идва от Алис, а не от друга личност, Боб

изпраща предизвикателство с голямото произволно число R_B. С това формира съобщение 2 и се изпраща като чист текст.

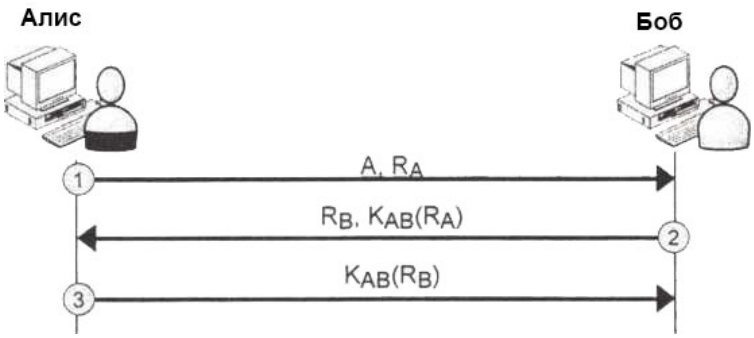
3. За да докаже самоличността си, Алис криптира произволното число със споделения таен ключ K_{AB}. Тя изпраща криптирания шифрован текст $K_{AB}(R_B)$ в съобщение 3 на Боб. На този етап Боб е сигурен, че съобщението е дошло от Алис, защото само Алис знае за тайния ключ K_{AB}. Освен това произволното число R_B е избрано от голям интервал (например 128-битово случайно число). Затова е невероятно някоя друга личност да е видяла съобщението в сесията.
4. Сега Боб е сигурен за проверката на самоличността на Алис. Но все още няма доказателство за самоличността Боб. За да провери дали Боб е този, за който се представя, Алис също изпраща случайно число R_A в чист текст на Боб в съобщение 4.
5. За да представи доказателство за самоличността си, Боб криптира слуайното число R_A със споделения ключ K_{AB}. Той изпраща криптирания текст $K_{AB}(R_A)$ в съобщение 5 на Алис. Когато Алис получи криптираното съобщение $K_{AB}(R_A)$, тя е сигурна, че взаимодейства с Боб.



фигура 45 *Автентикация чрез challenge-response протокол*

След като проверят самоличността си, ако Алис иска да продължи взаимодействието си с Боб, тя може да изпрати ключ за сесия K_S, криптиран с тайния ключ K_{AB}.

Горният процес е доста дълъг и съдържа някои допълнителни съобщения, които могат да се елиминират. На фигура 46 е илюстриран процеса на автентикация чрез ползване на challenge-response протокола само с три съобщения.



фигура 46 *Автентикация чрез challenge-response протокол с три съобщения*

Тристъпковият challenge-response протокол за автентикация е подобрение на оригиналния протокол. Но той има няколко дупки. Този тип протокол е предразположен към отразени атаки (reflection attacks). Разгледайте следния пример, за да разберете как отразената атака може да бъде започната в challenge-response протоколи. Нека приемем, че Боб е машина в банката, която позволява многобройни едновременни сесии с външни машини-касиери. Нарушител на име Мария започва отразена атака на Боб. Мария се представя за Алис и изпраща идентичността A (която е оригинално самоличността на Алис) и случайното число R_M на Боб. Боб както обикновено отговаря с изпращане на собственото си предизвикателство R_B и криптирания шифрован текст $K_{AB}(R_M)$. На този етап Мария е в затруднено положение, защото тя не знае за споделения таен ключ K_{AB}.

Сега Мария започва друга сесия с Боб чрез изпращане на R_B (взето от съобщение 2) като нейно предизвикателство. Боб както обикновено изпраща обратно отговор, който съдържа шифрован текст $K_{AB}(R_B)$, и собственото си предизвикателство. Мария получава липсващата информация и прекратява втората сесия. Тя завършва първата сесия чрез изпращане на съобщение с криптираното съобщение $K_{AB}(R_B)$. Тъй като Боб е убеден, че взаимодейства с Алис, той дава банковите подробности на Мария.

Учейки се от тристъпковия challenge-response протокол, има определени правила, които трябва да се следват при проектирането на всеки протокол за автентикация:

- Нека инициаторът докаже самоличността си преди отговарящия. В три-стъпковия challenge-response протокол Боб прави грешка, като предоставя ценна информация, преди да е сигурен за самоличността на Алис. Забележете, че при петстъпковия

challenge-response протокол инициа-торката Алис удостоверява самоличността си първа.

- Използвайте различни споделени тайни ключове, за да предоставите доказателство за самоличност. Това може да означава да използвате два споделени тайни ключа.
- Нека инициаторът и отговарящият използват различни набори от числа на предизвикателството. Например, едната страна може да използва четни числа, а другата - нечетни.

KDC механизъм за автентикация

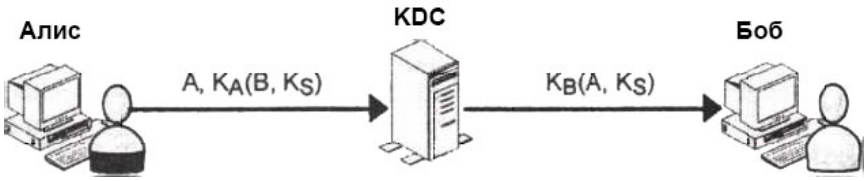
Споделянето на таен ключ с друга страна работеше доста добре, но вероятно не беше напълно успешно. То означаваше споделяне на N ключа за взаимодействие с N човека. Това можеше да доведе до проблем с управлението на ключовете, защото ще има няколко ключа.

Друг възприет подход е използването на доверен център за разпределение на ключове (key distribution center, KDC). При този метод всеки потребител съхранява своя ключ в KDC. KDC е отговорен за автентикацията и управлението на сесийни ключове.

Нека разгледаме действието на KDC чрез най-простия KDC протокол за автентикация - wide-mouth frog. Този протокол носи името си от псевдонима на своя създателя Michael Burrows.

Отново ще използваме примера с Алис и Боб, за да обясним идеята зад wide- mouth frog протокола (фигура 47). Следните стъпки изобразяват комуникацията между Алис и Боб през KDC:

1. Алис изпраща сесиен ключ K_S на KDC, заявявайки желанието си да започне връзка с Боб. Алис изпраща това съобщение като шифрован текст, криптиран с тайния ключ K_A, който тя вече е споделила с KDC.
2. KDC декриптира съобщението с тайния ключ на Алис K_A, за да вземе идентичността и сесийния ключ на Боб.
3. Тогава KDC създава ново съобщение със самоличността и сесийния ключ на Алис и го изпраща на Боб. Изпратеното съобщение е криптирано с тайния ключ на Боб K_B, който той вече е споделил с KDC.
4. Когато Боб декриптира съобщението с тайния си ключ K_B той знае, че Алис иска да комуникира с него, и знае още сесийния ключ, който тя иска да използва по време на комуникацията.



фигура 47 *Автентикация чрез wide- mouth frog протокол*

В горния процес се забелязва, че задачата за проверка на автентичността се управлява от KDC. KDC знае, че съобщението е дошло от Алис, защото тя криптира самоличността и сесийния си ключ със своя таен ключ, който вече е споделила с KDC. По подобен начин Боб знае, че съобщението е дошло от KDC, защото съобщението, което е получил, е криптирано с тайния ключ, който той вече е споделил с KDC.

Но протоколът wide-mouth frog е уязвим към повтарящи се атаки (replay attacks), при които валидно предаване на данни е злонамерено или измамно повтаряно или от създателя, или от нарушител, който пресреща данните и ги препредава.

За обяснение на повтарящата се атака, отново ще се използва примера с Алис, Боб и Мария. В този пример Боб е банкер, а Мария се опитва злонамерено да подслушва в мрежата. Мария, с намерението да открадне пари от банката, първо сключва сделка с Алис. Тя извършва някаква работа за Алис и след това я моли да преведе дължимото количество пари на нейната сметка.

Междувременно Мария пресреща съобщенията, които KDC изпраща на Боб. Тя също улавя заявката на Алис да преведе пари на сметката на Мария. След известно време Мария отново изпраща и двете съобщения на Боб. Този път Боб мисли, че Алис отново е сключила сделка с Мария, и затова превежда пари на сметката на Мария. С използването на повтарящи се съобщения Мария успява да се сдобие с пари от Боб.

Има няколко други протокола на KDC, които са по-сложни от wide-mouth frog протокола. Едни от тях са протоколът Needham-Schroeder и протоколът Otway-Rees за автентикация.

Kerberos V5

Kerberos е протокол за автентикация, който позволява на потребителите, взаимодействащи по мрежа, да удостоверят самоличността си.

Произход на Kerberos

Kerberos е протокол за автентикация в мрежа, който предоставя проверка на автентичност чрез независими услуги. Системи, инсталирани с Kerberos, изискват потребителят да напише паролата си само веднъж. Други услуги за проверка на автентичност (включително криптиране) за потребителя се управляват от самия Kerberos.

Kerberos беше първоначално проектиран в MIT през 1987 г. Оттогава този продукт е претърпял няколко промени и се е развил до стандартизиран продукт, поддържан от най-често използваните операционни системи и приложения. Kerberos все още подлежи на положителни разработки с новите издания, които излизат приблизително два пъти годишно. Версия 5 се използва от Microsoft за автентикация в Windows.

Обикновено Kerberos се изпълнява в програми на ниво приложения като например Telnet, FTP, rsh, rep, rlogin и ssh. Той може да се вгради в други програми. Например, той може да се инсталира на машина, за да се приложат свойствата на автентикацията на Kerberos. Програми, осъществяващи Kerberos, или машини инсталирани с Kerberos софтуер, се наричат укрепени или керберизирани програми или машини.

Услуги на Kerberos

Kerberos потвърждава самоличността на потребител или на мрежова услуга, използва концепцията на споделения таен ключ. Той позволява на клиенти и сървъри в мрежа взаимно да установят самоличностите си, без да разкриват пароли. Чрез използването на криптографски методи той също подсигурява поддържането на поверителността и секретността на информацията, която се предава.

Kerberos основно действа чрез използване на билети (tickets). Билетът е последователност от няколкостотин байта, които лесно могат да се обменят виртуално по всички мрежови протоколи. Билетите подсигуряват,

че процесът, използващ Kerberos, може да потвърди самоличността на потребители и ресурси по мрежата.

В Kerberos V5, автентикацията на парола се извършва в централно място за всички машини в укрепена среда. Тя не се извършва в системи от страна на потребителите. Системите от страна на потребителите не участват в автентикацията на пароли и в процеса на поддръжка.

Обобщение:

- Kerberos предоставя автентикация и сигурни услуги за комуникация на основни единици в мрежа.
- Той генерира тайни ключове за заявяващите и предоставя механизъм за сигурно предаване на ключовете по мрежата.

Kerberos области

Kerberos разделя мрежата на домейни на сигурност, наречени области. Всяка област има сървър за автентикация, който осъществява своя собствена политика за автентикация. Поради съществуването на различни области в мрежата, организациите, които използват Kerberos, могат да поддържат различни нива на сигурност с различни политики на сигурността за различните отдели на организацията. Една област може да приема автентикация от другите области, ако политиката на сигурността, дефинирана в нея, го позволява. Но тя може да откаже проверка на автентичност, ако политиката изисква повторен такъв процес.

Политиките, дефинирани в Kerberos, са йерархични. Това означава, че областта може да има дъщерни области под себе си. Йерархичната структура позволява дъщерните области, които нямат права за директна проверка на автентичност, да споделят информация за автентикацията с родителските области. Например потребител в организация от една област може да иска да се свърже с компютър от друга област. Потребителят в този случай няма нужда да минава през повторен процес на автентикация, ако компютърът, с който иска да се свърже, е в родителската област.

Компоненти, включени в Kerberos V5

Kerberos включва три услуги в процеса си освен клиентските работни станции:

- *Authentication Service (AS)* - услуга за автентикация. Проверява автентичността на потребителите по време на регистрация.
- *Ticket-Granting Service (TGS)* - услуга за издаване на билети. Издава билети, които са доказателство за самоличност.
- *Application Server* - сървър за приложения. Предоставя услуги на клиента.

Тези ключови компоненти работят заедно, за да предоставят услуги за автентикация на клиента. Kerberos V5 използва друг важен компонент освен изброените по-горе. На първо място осъществява автентикацията през центъра за разпределение на ключове (KDC). В предходната секция дискутирахме основните функции на KDC в процеса на автентикацията. Основната роля на KDC в Kerberos е да осъществи съвместно услугите за автентикация и услугите за издаване на билети за всички машини в областта. Той споделя постоянен таен ключ за всяка основна единица (потребители и услуги в мрежата). Той поддържа базата от данни, в която се съхраняват детайли за всички основни единици в областта. Заради това, KDC също се нарича *база от данни на Kerberos*.

Процесът на автентикация в Kerberos включва следните термини:

- *Сесиен ключ (Session key)*. KDC по случаен начин генерира временен таен сесиен ключ, който трябва да се сподели между KDC и клиента. Сесийният ключ съществува дотогава, докато придружаващият го билет е жив. Основната цел на сесийния ключ е да проверява автентичността на основните единици по няколко пъти по време на съществуването на билета, за да ограничи използването на споделения таен ключ (който в действителност е хеш стойността на паролата) по мрежата.
- *Билет (Ticket)*. Kerberos използва билети, за да провери автентичността на основни единици по мрежата. Билетите са криптирани записи, съдържащи сесийния ключ, самоличността на потребителя и идентификатора на услугата, както и IP адреса на клиента. Част от информацията, съдържаща се в билета, е криптирана със споделения таен ключ, който е известен само на KDC и на основната единица. Билетът е придружен и от копие на сесийния ключ, което е криптирано със споделения таен ключ. Способността на основните единици правилно да декриптират

съответните части на билета определя автентичността им по мрежата.

- *Удостоверение за самоличност (Credential)*. Комбинацията от сесийния ключ и билета е известна като удостоверение за самоличност.

Услугата за автентикация издава тайни сесийни ключове и удостоверения за самоличност на базата на потребителската парола и ключ за криптиране. Тя може също да издаде пропуски, въпреки че това е функция на услугата за издаване на билети. Всеки от билетите, издаден на клиенти, е билет за индивидуална керберизирана услуга, която клиентът иска да използва.

Процес на автентикация в Kerberos

Ще разясним процеса на автентикация с помощта на примера за Алис и Боб: Следват необходимите стъпки, когато Алис (потребител, използващ укрепена машина) иска да комуникира с Боб (сървър за приложения) ():

1. Алис, използваща укрепена машина, написва името си за автентикация. Машината изпраща идентичността си A към KDC като чист текст(машината не пита за парола).
2. Услугата на KDC за автентикация отговаря с криптирано удостоверение за самоличност в тайния ключ на Алис. Криптираното удостоверение за самоличност включва сесиен ключ K_S и билет $K_{TGS}(A, K_S)$, който в действителност е за услугата за издаване на билети на KDC. (Алис вече е споделила тайния си ключ с KDC.). На този етап с помощта на тайния ключ на Алис съобщението се декриптира отново, за да бъде получен сесийният ключ и TGS билетът. Тъй като Алис може да декриптира съобщението с тайния си ключ, системата е сигурна за нейната самоличност. Затова само след успешно декриптиране работната станция пита Алис за парола.
3. След като Алис успешно се идентифицира в системата, тя изпраща съобщение на KDC отново със заявка за издаване на TGS билет за Боб - сървъра за приложения. Компонентите на това съобщение са $K_{TGS}(A, K_S)$, B, $K_S(t)$. В тези компоненти t представя маркер на времето. KDC се опитва да декриптира съобщението с копието на тайния ключ на Алис, което има. Ако успее да го направи и ако маркерът за време t е скорошен, KDC е сигурен, че Алис е същата личност, за която се представя.

4. След това услугата за издаване на билети на KDC отговаря чрез създаване на сесиен ключ $К_{AB}$, който Алис може да използва с Боб. TGS изпраща две версии на сесийния ключ - $K_S(A, К_{AB})$ и $К_B(A, К_{AB})$. Първият ключ е криптиран с тайния ключ, който KDC е издал на Алис, така че Алис да може да прочете съобщението. Вторият е криптиран с тайния ключ на Боб, така че той да може да го декриптира. Алис не може да разбере втората версия, защото тя не знае тайния ключ, който Боб е споделил с KDC.
5. Сега Алис изпраща копие на сесийния ключ $К_{AB}$, криптиран в тайния ключ на Боб. Това съобщение отново е придружено от същия маркер за време t.
6. Отговорът на Боб $К_{AB}(t+1)$ е доказателство от Боб, че Алис говори действително с Боб и с никой друг.

Цифрови подписи

Законността на правни, финансови и други документи се определя от наличието на ръкописен подпис на упълномощените страни на документите. В повечето случаи документите трябва да се представят в оригиналната им форма. Обикновено не се приемат фотокопия на документите.

В наши дни бизнес комуникациите стават онлайн. Компютъризирани системи за съобщения почти са изместили обмена на физически документи. Но как може някой да е сигурен за автентичността на документите, които се обменят онлайн?

Не е лесно да се измисли заместител на ръкописните подписи. В основни линии трябва да се създаде такава система, чрез която един потребител да може да изпраща цифрово съобщение на друг. Една подобна система трябва да отговаря на следните изисквания:

- *Получателят трябва да може да потвърди самоличността на подателя*. Институции, които предоставят финансови услуги, не биха могли да работят онлайн, ако това условие не е изпълнено. Например клиент може да нареди на банката си да впише като дълг в сметката му определена сума пари, защото е направил покупки. На този етап, преди банката да процедира, тя трябва да е сигурна за самоличността на клиента.
- *Подателят трябва да не може да отрече съобщението, което е изпратил*. Разгледайте примера с банката отново. Клиент може

да поръча на банката да купи дялове като част от инвестиционната програма на банката. Скоро след това цената на дяловете пада рязко. Нечестен клиент се отрича от съгласието си и казва, че никога не е изпращал подобно съобщение.

- *Получателят трябва да не е способен да променя съобщението си по-късно*. Връщайки се към примера с банката, нека приемем, че скоро след като сделката между клиента и банката е завършила, цената на дяловете се качва. Банката може да се опита да създаде подписан договор, който гласи, че се купува по-скоро един дял, отколкото сто.

Има много методи за създаване на цифрови подписи. Няколко от тях са с използване на тайни и публични ключове.

Подписи с таен ключ

В този метод централният орган (central authority, CA), който всички познават и на който всички вярват, играе важна роля. Отсега нататък в обясненията централния орган ще се означава с CA. Всяка страна споделя таен ключ с CA, така че тя да може да използва цифрови подписи на документите, които се предават. Тези тайни ключове са тайна между заинтересуваната страна и CA.

Когато някоя страна иска да изпрати съобщение на друга страна, тя се свързва с CA. Отново се връщаме на примера с Алис и Боб. Да приемем, че Алис иска да изпрати съобщение с цифров подпис на Боб. За целта Алис изпраща съобщение до CA, което съдържа подробности като име на страната, на която иска да изпрати съобщение (в нашия случай Боб), произволно число, маркер за време и съобщението. Всички тези подробности са криптирани с тайния ключ на Алис. CA знае, че съобщението идва от Алис, тъй като е криптирано с тайния ключ, който Алис е споделила с него. CA декриптира съобщението и изпраща съобщение на Боб. Съобщението, изпратено на Боб от CA, съдържа криптиран цифров подпис, който също съдържа маркер за време.

Този метод не се използва често в наши дни заради споровете, които е предизвикал в миналото. Въпреки че споровете са разрешени по-късно, той все още не е много търсен избор. Какво става, ако Алис отрече съобщението, което е изпратила на CA? В повечето случаи всички страни биха се съдили. Съдът моли CA да докаже, че Алис е изпратила съобщението. CA показва

съобщението, което е изпратила Алис. Съобщението, криптирано с тайния ключ, който е споделен само между Алис и CA, доказва, че Алис греши.

Подписи с публичен ключ

Един недостатък на цифровите подписи с тайни ключове е, че всяка страна трябва да вярва на CA. Още повече, всяко съобщение трябва да премине през CA, дори и страната да не иска да стане така. CA може да чете всички поверителни взаимодействия между двете страни. Най-вече централизираният доверен орган е правителството, банките или адвокатите. Но те също не получават пълна поверителност от хората. По-скоро е трудно за всеки човек да вярва на някой орган безусловно. Затова неофициално комбинирано споразумение между двете комуникиращи страни заключава, че е най-добре да не се намесват външни органи в цифровото подписване на документи.

За щастие е открит друг метод за създаване на цифрови подписи. Той включва използването на криптография с публичен ключ. Всъщност, този метод беше много търсен избор. Най-простият начин за създаване на цифрови подписи с използване на криптография с публични ключове може да включва създателят на съобщението да криптира подписа си с таен ключ, изолирайки подписа в оригиналното съобщение. Всеки с публичния ключ на създателя може да декриптира съобщението до оригиналното съобщение. Автентичността на подписа се потвърждава, когато декриптираното съобщение съвпадне с оригиналното, защото само някой с частен ключ може да създаде подписа.

Но този метод също има своите пропуски. Криптирането на данните за предоставяне на цифров подпис не е приемливо поради следните причини:

- Криптираният подпис, съдържан в съобщението, обикновено е със същия размер (ако не и по дълъг) като самото съобщение. Затова пълното съобщение (съобщение + цифров подпис) удвоява размера си, заема много място за съхранение и създава по-голям трафик на данни. Предаването на това съобщение по мрежата включва ненужно натоварване.
- Криптирането с публичен ключ е сложно и поради това малко бавно. То също така причинява тежко натоварване в изчисленията на процесора. Затова, при предаването на такива

съобщения с цифров подпис, работата на мрежата, както и на компютъра, забележимо се влошават.

- Подобни съобщения са уязвими към атаки с криптоанализ, тъй като атакуващите знаят определени части от чистия текст на съобщението предварително. Пример за позната част на съобщението е темата (subject) на e-mail съобщението, която е част от заглавната част (header) на съобщението.

Като погледнем горната част, прикачена към цифровите подписи, създадени с използване на криптография с прости публични ключове, изглежда, че алгоритмите за цифрови подписи са по-добра опция. Тези алгоритми също са базирани на криптографията с публични ключове, но използват по-ефикасни методи като обобщения (digests) на съобщения. Съобщения, които изолират цифровите подписи, базирани на алгоритми, използват частния ключ на създателя, за да създадат цифров "отпечатък" на данните. Поради механизма с обобщение на съобщения, размерът на подписа обикновено е по-малък от този на цифровите подписи, създавани по-рано.

Освен това цифровите подписи, използващи обобщения на съобщения, причиняват относително по-малко натоварване на процесорите, създават по-малко трафик и също генерират криптиран текст, който е относително по-малък на размер.

Най-широко разпространените алгоритми за цифрови подписи са процесът за цифрови подписи RSA и алгоритъмът за цифрови подписи DSA (Digital Signature Algorithm).

Обобщения на съобщения

Обобщенията на съобщения генерират цифрови обобщения на информация (документи и файлове, предавани онлайн). Обобщенията на съобщения са в по-голямата си част базирани на хеш функции. Хеш функция е математическо изчисление, приложено на съобщението, за да се генерира малък низ, наречен обобщение. Обобщението представлява пълния файл или документ.

Дължината на едно обобщение на съобщение обикновено варира между 128 и 160 бита. За всеки документ се създава уникално обобщение на съобщението. Идентичните документи имат еднакви обобщения, но дори и

един бит от текст; да е различен, полученото обобщение на съобщението е различно. Уникалността на обобщенията на съобщения зависи от създателя. Това означава, че идентични документи също могат да имат различни обобщения. Всичко зависи от функциите за обобщения на съобщения, използвани за създаването им.

Също така е невъзможно да се създадат идентични обобщения на две различни съобщения.

Всяко обобщение на съобщение, създадено чрез частен ключ, е уникално за създателя си и може да се декриптира само със съответния публичен ключ.

Обобщенията на съобщения се използват заедно с технологията на публичните ключове, за да се създават цифрови подписи. Те също се използват и като цифрови “отпечатъци”, за да поддържат автентичността, целостта и признаването на съобщения.

Друга важна функция на обобщенията на съобщения е да потвърдят целостта на цифрово подписани електронни файлове и документи. Тази концепция е подробно обяснена в секцията за RSA цифровите подписи по-долу.

Два често използвани алгоритъма за обобщения на съобщения са MD5, 128-битовс обобщение, разработено от RSA Data Security Inc., и SHA, 160-битово обобщение на съобщение, разработено от Националната агенция за сигурност в САЩ.

Обобщенията на съобщения с функция HMAC (Hashed Message Authenticatication Code) е друг механизъм за подсигуряване на целостта на документи.

Функцията HMAC

Функцията HMAC (Hashed Message Authentication Code) се използва за автентикция на съобщения, предавани по мрежа. HMAC е описана в RFC 2104 на Network Working Group на Internet Engineering Task force (IETF). HMAC MD5 също използва стандартните функции за обобщения на съобщения, MD5 and SHA.

HMAC е широко използвана от Интернет технологии, като например TLS и IPSec протоколите, като средство за потвърждаване на целостта на

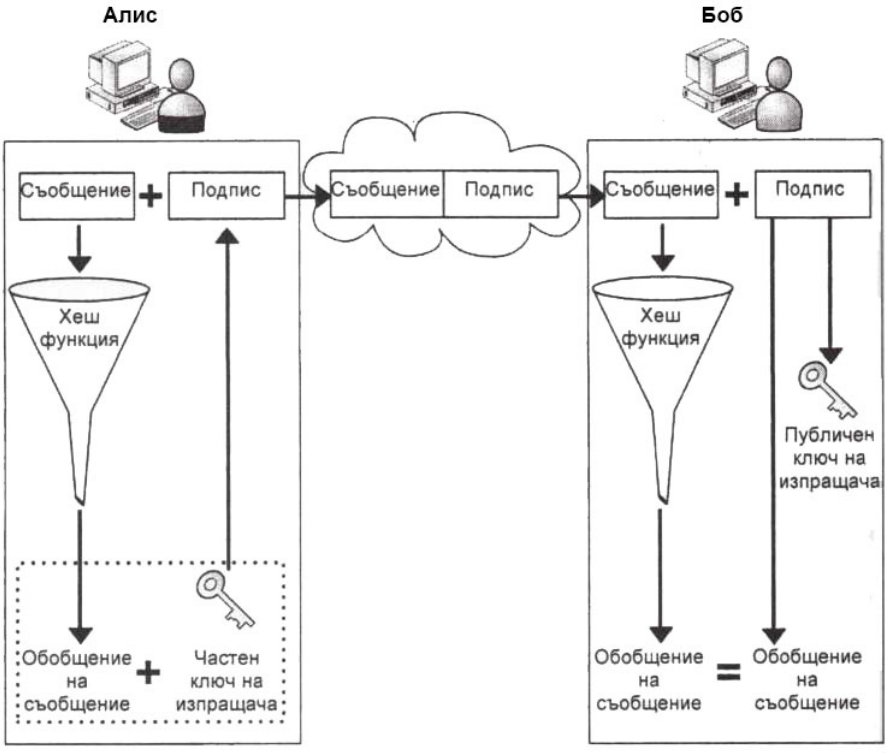
данни, предавани по несигурна мрежа. HMAC действа чрез създаване на обобщение на съобщение за всеки блок от данни. Тя използва случаен симетричен ключ, за да криптира обобщенията. Тайният ключ е споделен между заинтересуваните страни, както в случая на криптирането със симетричен ключ. Криптираните данни се декриптират със споделения таен ключ.

Ефективността на HMAC криптираните данни зависи от силата на обобщението на съобщение, използвано в криптирането на данни, и от това колко сигурно данните са предавани по мрежата. Трудно е за нарушител да се справи с предавани HMAC данни, защото той не знае тайния ключ. За разлика от цифровите подписи HMAC не изисква комуникиращите страни да имат публични и частни ключове.

След като разгледахме обобщенията на съобщения, нека да разгледаме алгоритмите, които ги използват за създаване на цифрови подписи.

RSA цифрови подписи

RSA цифровите подписи използват частен ключ, за да криптират обобщението на съобщение, което формира цифровия подпис. Тогава цифровият подпис се прикачва към оригиналното съобщение.



фигура 48 *Процеса на създаване на RSA цифровия подпис*

Вече разгледахме как цифровите подписи се криптират и прикачват към съобщенията. Но как може получателят да е сигурен за целостта на прикачения цифров подпис? Има възможност съобщението да е подслушано по мрежата и да е направена промяна на самия цифров подпис. За да удостовери съдържанието на цифрово подписано съобщение, получателят трябва да генерира ново обобщение на съобщение от съобщението, което е получил. Той прави това, като декриптира полученото съобщение с публичния ключ на създателя. След това сравнява декриптираното обобщение с новогенерираното обобщение. Ако двете версии си съвпадат, целостта на съобщението е потвърдена. Автентичността на създателя също се потвърждава, защото публичният ключ може да декриптира само това съобщение, което е криптирано със съответния частен ключ от двойката.

Алгоритъм за цифрови подписи DSA

Друг механизъм за създаване на цифрови подписи е Digital Signature Algorithm (DSA). DSA е механизъм, дефиниран в стандарта DSS (Digital Signature Security Standard), представен от Националната агенция за сигурност в САЩ. DSS се приема за стандарт за цифрови подписи от американското правителство.

Функционалността на DSA е подобна на RSA. Но за разлика от RSA, DSA не криптира обобщенията на съобщения с частен ключ. Вместо това DSA използва математически функции, за да генерира цифров подпис. Тези цифрови подписи се създават чрез използване на две 160-битови числа, произхождащи от обобщението на съобщение и частния ключ. DSA използва публичен ключ, за да удостовери подписа. Но този процес на потвърждаване е много по-сложен от RSA.

Имейл удостоверяване

Имейл удостоверяване (или валидация) е набор от технологии, които доставчиците на имейл услуги използват, за да потвърдят идентичността на имейл сървъра, изпратил дадено съобщение. Оригиналният протокол, използван за обмен на съобщения SMTP (Simple Mail Transfer Protocol) няма подобна функционалност. Създаден е през далечната 1982 г.

Две основни технологии се използват за имейл удостоверяване:

- **SPF** – Sender Policy Framework;

- **DKIM** – DomainKeys Identified Mail.

Как работи имейл удостоверяването?

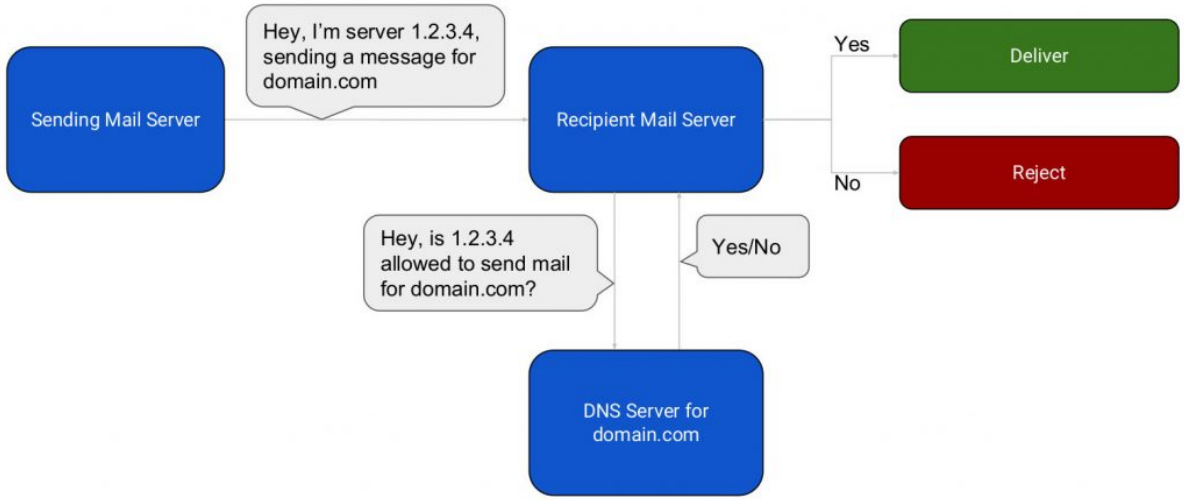
SPF и **DKIM** използват различен подход да удостоверят дадено съобщение. И двете използват **DNS** записи, но те съдържат различна информация. Когато имейл сървър получи съобщение, той проверява тези записи и на базата на резултатите, които получи, определя дали съобщението е пратено от легитимен сървър или не.

SPF

SPF DNS записът съдържа информация кои имейл сървъри (**SMTP**) могат да изпращат писма за дадено домейн име.

```
v=spf1 a mx include:smtp-spf.someserver.com ~all
```

При **SPF** защита, могат да се заложат различни **SMTP** сървъри в DNS записа без да се налага да се прави нещо по-специално, на който и да е от тези сървъри. Последната част от записа (~all) указва по какъв начин трябва да се третира съобщение, получено от сървър, който не е споменат в записа. В примерния запис, "~all" означава "**SOFTFAIL**". С други думи, подобно съобщение ще бъде маркирано като неудостоверено, но няма да бъде отхвърлено. За отхвърлянето му, може да се използва „-all".



фигура 49 *Схема на процес по SPF валидация*

SPF защитата работи само, ако получаващият сървър прави проверка за **SPF имейл удостоверяване**. Когато сървърът получи съобщение, проверява дали IP адресът на изпращащия сървър е изброен в **SPF DNS** записа на домейн името на изпращача. Според резултата и настройката как

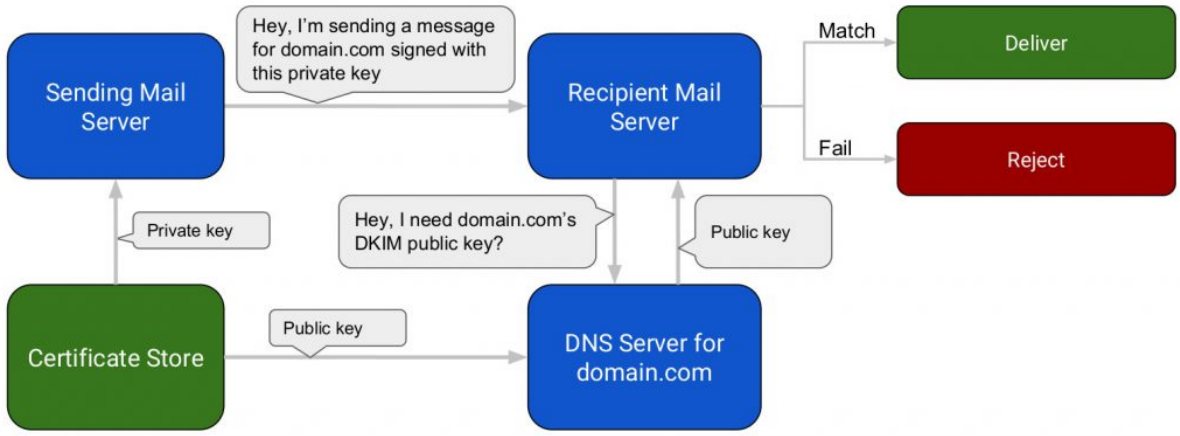
да се третира подобно съобщение (~all, например), сървърът решава дали съобщението да бъде доставено нормално, отхвърлено, или доставено и маркирано като спам.

DKIM / DomainKeys

DKIM (DomainKeys Identified Mail) защитата използва публичен криптографски ключ, с който съобщенията се подписват. Всеки сървър, оторизиран да изпраща съобщения, трябва да подписва съобщенията с частен ключ (private key). Този ключ има уникален съответстващ публичен ключ (public key), който се публикува в DKIM DNS записа. Имейл сървърите, които получат дадено съобщение, използват този публичен ключ да проверят автентичността на съобщението (тъй като само оторизирани сървъри имат правилния частен ключ от двойката ключове).

Примерен DKIM запис:

```
v=DKIM1; k=rsa;
p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDTYf0C/D2g0vFGjYWb5gzZ2WDnBH
iq++iHkCyQ1upCSokENuDAgRfAiWrW1M1Ge6dbZoI5RPzChmKe7PMqZf7fzj0+dvn6VP
//r/+cZd6nmMh65cN/iwwl7ncP6rngI8B4cfmgPfjU0eY46F511mThvAQ4TLvj7Han5q
MZrG5FzwIDAQAB
```



фигура 50 *Схема на процес по DKIM валидация*

Използването на DKIM защита предполага, че сървърът, който получи съобщение, ще провери неговата автентичност. Ако подобна проверка е пусната, получаващият сървър ще провери т.нар. имейл хедъри на съобщението и ще ги сравни с публичния ключ в DKIM DNS записа за домейна. Ако съобщението не мине тази проверка успешно, то ще бъде третирано според настройките на сървъра за подобен случай.

DMARC (Domain-based Message Authentication, Reporting and Conformance)

Това е проект за противодействие на спам и измамни имейл съобщения. В разработката и имплементирането на технологията вземат участие компании като AOL, Google, Hotmail, Yahoo! Mail, PayPal, Facebook, LinkedIn и други.

Проектът е представен за пръв път през януари 2012 г. В рамките на следващата една година, DMARC се разпростира и имплементира до 60% от имейл услугите предлагащи се в интернет.

DMARC е нова технология за защита от спам, която използва за основа вече съществуващите механизми за валидиране на имейл съобщенията - SPF и DKIM. DMARC надгражда тази основа с две значителни възможности:

- обратна връзка под формата на доклад, генериран от получателя, съдържащ подробности за полученото имейл съобщение;
- контрол върху съдбата на невалидираните имейл съобщения - да се маркират като спам или директно да се отхвърлят още преди да стигнат при получателя.

С докладите (DMARC reports), генерирани и изпращани от получателите, може да се установи дали домейн името се използва за изпращане на измамни имейл съобщения. Контролът, прилаган чрез предварително посочени политики, указва за всеки имейл, който не е преминал успешно валидацията, дали да бъде маркиран като спам от получателя или да бъде отхвърлен, без да достига до входящата поща.

При активирането на DMARC за даден домейн първо е необходимо да се добави SPF и DKIM имейл идентификация. След това, се добавя TXT запис в DNS зоната за домейна. Резултатът е активиран DMARC и получаване на доклади.

Примерен DMARC TXT запис:

```
_dmarc.mydomain.com "v=DMARC1; p=none; rua=mailto:dmarcreports@mydomain.com;"
```

```
v= версията на DMARC

p= политиката за контрол на невалидираните имейли

rua= имейл адресът, на който ще получавате докладите
```

Обикновено за създаването на пълен и коректен DMARC TXT запис за домейн, се използват инструменти като DMARC Record Assistant.

DMARC доклади ще бъдат изпращани от получателите, само ако към техния входящ имейл сървър също е имплементиран DMARC.

DMARC политики

В DMARC записа се съдържат няколко DMARC тага (параметри). Един от тях (p=) определя какво ще се случи на всяко писмо, което не е преминало успешно DMARC валидирането при получателя.

Изборът е между три възможни политики (инструкции):

- **p=none** - режим мониторинг; върху невалидираните имейли не се прилага никакво действие;
- **p=quarantine** - невалидираните имейли да се маркират като измамни (попадат в папка Junk, Spam);
- **p=reject** - невалидираните имейли се блокират и отхвърлят преди да попаднат във входящата кутия на получателя .

DMARC политиката, зададена в TXT записа, ще се приложи, когато някой получи имейл, в който вашият домейн е посочен за изпращач, но реално имейлът е изпратен от неоторизиран мейл сървър.

На база получените доклади и резултатите от тях, DMARC защитата може да премине към по-строга политика.

DMARC доклади (рапорти)

DMARC докладите представляват текстови файлове във формат *.xml*, които се генерират и изпращат от мейл сървъра на получателя.

За обработка и анализ на тези доклади е необходим подходящ инструмент, който да преработи .xml данните в разбираем формат, например: DMARC XML-to-Human Converter.