Седма глава

АТАКИ В КОМУНИКАЦИОННИ МРЕЖИ

7.1 ВИДОВЕ АТАКИ

Целта на интрузията може да бъде неоторизирано придобиване, подмяна или унищожаване на информация, както и неоторизирано придобиване на права за ползване на комуникационни ресурси.

Обектът, към който е насочена атаката ще наричаме **обект на интрузия**, а провеждащият атаката – **интрузионен** агент или **интрудер**[1].

Атаките провеждани в мрежите са от два типа **активни и пасивни**.

Пасивни са атаките, които са свързани с подслушване на пакети (packet sniffing) и анализ на трафика (traffic analysis).

Целта на **пасивните атаки** е събирането на информация за атакувания обект. Събраната информация се използва за анализиране на обекта и откриване на слабите страни в неговата защита. Атаките от този тип са трудни за откриване и идентифициране. Тяхното действие е аналогично на действието на програмите за мониторинг на мрежата и записване на трафика. Резултатите от анализите на събраните данни се използват за провеждане (информационно осигуряване) на активни атаки.

Активни са атаките, свързани с подмяна на права или унищожаване на ресурси, както и с подмяна на данни или изтриване на данни.

Активните и пасивните атаки са насочени или към операционните системи на крайните устройства и техните комуникационни стекове или към мрежовите устройства. В учебника са разгледани техническите аспекти за реализиране на различни атаки[2], насочени основно към комуникационните протоколи, докато атаките насочени към операционните системи са представени информативно.

7.2 АТАКИ, НАСОЧЕНИ КЪМ ПОТРЕБИТЕЛЯ И ПЕРАЦИОННАТА СИСТЕМА

7.2.1 Социален Инженеринг (Social Engineering).

Социалният инженеринг е метод за неоторизирано придобиване на информационни ресурси и/или потребителски права без използване на технически средства. Целта е да се изучат навиците на определен човек или

---

[1] В литературата много често вместо понятието „интрудер” се използва понятието „хакер”или „кракер”. Тяхната интерпретация е различна при различни автори и затова в тази книга ще използваме по-общото понятие.

[2] По-подробно описание на видовете атаки с отчитане и на човешкия фактор и природни влияния може да се намери на адрес http://www.all.net/journal/ntb/cause-and-effect.html.

група хора, политиката за сигурност на организацията или институцията и параметрите на атакуван обект. След това да се анализира до каква степен се следват предписанията за сигурност на работното място и има ли слабо място, което може да бъде атакувано (използвано).

Методите за провеждане на социален инженеринг са:

1. Придобиване на права и/или данни, чрез фалшиви правомощия;
2. Придобиване на права и/или данни посредством ползване на фалшива самоличност;
3. Предлагане на удовлетворяване на интереси срещу неоторизирано предоставяне на права или данни.

На социалният инженеринг се противодейства посредством:

- провеждане на непрекъсната политика за подобряване на сигурността;
- анализ на степените на риск по отношение на отделните звена и личности в съответната организация;
- унищожаване на хартиените отпадъци и електронни документи със специализирани машини;
- мотивация на хората по отношение на секретността;
- ограничаване на точките за изтичане на информация;
- провеждане на непрекъснато образование по отношение на политиката на сигурност;
- създаване на финансова заинтересованост от наличието на високо ниво на сигурността.

7.2.2 «Троянски кон»

„Троянският кон”[3] е програмно средство за интрузия, имащо за цел да изпълнява действия, скрити за потребителя на мрежата или крайната система. Основната цел на този вид програми е да отварят входни точки (портове) в операционната система, с което да осигуряват ресурси за последващи атаки.

Разновидностите на “троянските коне” според начина на програмна реализация са:

- самостоятелно работеща програма;
- вмъкване (добавяне, имплементиране) в програма на допълнителен или модифициран сорс-код, реализиращ функции, свързани с провеждане на определена интрузионна дейност.
- троянски двоичен файл – предварително компилиран системен файл с допълнително вградени функции. След успешна интрузия в дадена система с този файл се подменя определен системен файл[4] с цел използването му впоследствие.

---

[3] Названието е алюзия на старогръцката легенда за дървения кон, чрез който с измама е превзета Троя.

[4] Системни се наричат файлове, инсталирани от операционната система, които се ползват за реализиране на различни нейни функции.

Пример за такава програма е прихващане на парола. Тя се стартира под формата на екран за прихващане на пароли (*password grabber*), който има външен вид, аналогичен на екрана, ползван от атакуваната програма за нормално въвеждане на парола. След въвеждане на потребителското име и паролата в предоставения интерфейс, следва записване на данните във файл, показване на фалшиво съобщение за неуспешна операция и активиране на нормалната програма. Впоследствие файлът с прихванатата парола може да се изпрати до определен адрес в Internet с цел потребителското име и паролата да се използват за интрузия.

Заразяването с „троянски коне" става чрез използване на изпълними файлове, usenet съобщения, email спам, фалшиви корекции (updates) на сорс код, фалшиви програми за тестване на сигурността, скрийнсейвъри, нелицензиран софтуер и др.

Срещу атаки от тип "троянски кон" може да се противодейства като се спазват някои от следните правила:

- Никога да не се допуска изпълнение на програма, получена от недостоверен източник.
- Преди да се стартира дадена програма, трябва да се знае какво прави.
- Всяка получена програма да се сканира с актуален защитен софтуер.
- При използване на чужд сорс код е необходимо да се анализира и провери действителната му функционалност преди компилация.
- Ако свалените файл има обявена контролна сума[5] или прикрепен електронен подпис задължително трябва да се провери.

7.2.3 Вируси

Компютърният вирус е *програма*, която не може да функционира независимо. Вирусът се прикрепя като част от друга програма и променя функционалността й.

Вирусите могат да причинят следните нарушения:

- промяна на функционалност на програма;
- унищожаване на данни;
- блокиране на работата на ресурс или услуга;
- подмяна на настройки на операционна система.

Вирусите се разпространяват чрез заразени файлове[6], предавани чрез дискети, електронна поща, файлове, копирани от споделени директории, файлове "изтеглени" от Internet (download).

---

[5] Контролните суми представляват хеш-стойности на файловете. Начинът им за използване е описан в главите, свързани с криптография.

[6] Файлове, в който вирусът е разположен от създателите му.

Вирусите се обозначават с имена. Имената им са два типа – псевдоними (нарицателни, жаргонни имена) и класификационни имена, давани от различни изследователски организации и производители на антивирусен софтуер.

Неутрализирането на вирусите се извършва по два начина:

- директно откриване на вируса и премахването му с редактор или
- използване на антивирусна програма.

Директното премахване на вируса изисква добро познаване на заразената операционна система и/или продукт и не е обект на настоящия учебник.

Антивирусните програми са два типа:

- Програми, работещи чрез **сигнатурен анализ** или
- Програми, работещи чрез **имунни системи**.

Сигнатурният анализ е основан на търсене на предварително зададени символни поредици в изследван файл. Всеки файл е поредица от символи (байтове).

Когато се създава една програма (независимо нормална или с вирус) тя се програмира на език от високо ниво. След това чрез компилатор или транслатор тя се преобразува в изпълним код, който представлява машинни инструкции, подавани към процесора. Всяка машинна инструкция може да се разглежда като поредица от байтове. Програмата–вирус е именно такава поредица, вградена в байтовете на основната програма носител.

Сигнатурният анализ търси в изследвания файл предварително известни поредици от байтове (сигнатури, символи). Сигнатурите са записани в речници (бази данни или текстови файлове). При откриване на съвпадение се прави опит за премахване на вируса от заразения файл (дезинфекциране), изолиране на заразения файл в друга директория (или преименуване) или изтриване на заразения файл. Когато този файл е системен и се изтрие, той може да наруши работоспособността на операционната система. Ако пък е програма, изтриването й ще доведе до загуба на функционалност и необходимост от последващо инсталиране.

Имунните системи представляват софтуер, който притежава сигнатури на вируси, и има възможност да изгражда допълнителни хипотези за наличие на вируси в сканираните файлове. При откриване на вирусна сигнатура имунните системи реагират аналогично на антивирусните програми. Могат да открият и сигнатури, сходни с предварително дефинираните или притежаващи инструкции, които противоречат на предназначението на програмата. Тогава изследваните файлове се поставят под карантина – преместват се в определена директория и тяхното стартиране не се допуска.

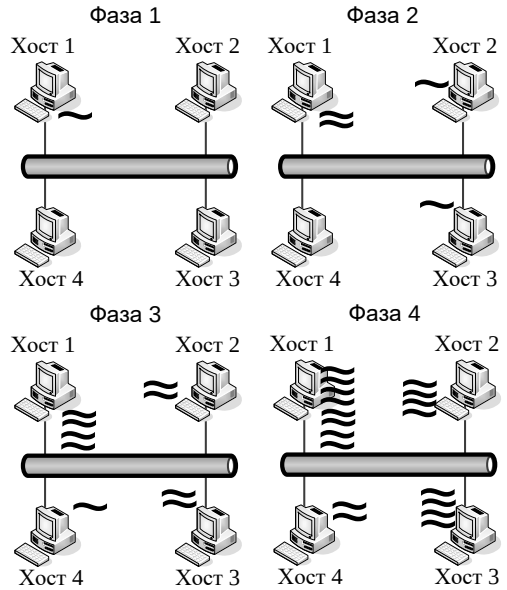
7.2.4 «Червей» (Worm)

«Червеят» е самостоятелно функционираща и независима програма, която е проектирана да се размножава (разпространява) в заразената система или да търси други системи в рамките на достъпните мрежи с цел тяхното заразяване [38].

Основната разлика между вирусите и «червеите» е функционирането на червея като самостоятелна програма. Тази разлика бързо се стопява и се получават нови хибриди, обединяващи и двата начина на съществуване. Пример за това е вирусът Мелиса, разработен през 1999 г. Разпространява се посредством прикрепен файл, които започва своето възпроизвеждане и саморазпространяване след заразяване на операционната система.

Саморазпространението на един червей се реализира чрез използване на инсталирани комуникационни услуги на заразената машина. Такива като електронна поща, услуга за трансфер на файлове, услуга за отдалечен достъп до друга машина и др.

Възможен сценарии за разпространение на един червей е дадена на фиг.7.1.



Фиг. 7. 1 Разпространение на вирус в мрежа

Вирусът заразява в началото хост 1, при което се активира и започва едновременно да се размножава и разпространява в рамките на локалната мрежа. В процеса на разпространение той търси хостове, които до момента не са заразени. В случая, първо заразява хостове 2 и 3 (фаза 2). Във фаза 3 са заразени всички компютри в мрежата. Във фаза 4 се наблюдава размножаване на вируса в рамките на локалната мрежа. В процеса на размножаване програмата-вирус започва да изчерпва ресурсите (например по отношение на памет) на операционните системи на отделните хостове и те последователно губят работоспособността си. След време се нарушава работоспособността на цялата локална мрежа.

7.2.5 Софтуер-шпионин[7]

Шпионският софтуер има за цел събирането на персонална информация за дейността на потребител (включително и какво купува през Internet) или да доставя рекламни материали. Тези програми се наричат **spyware**.

Spyware е софтуерен продукт, който се вмъква в операционната система и има за цел прихващането на въведената информация от клавиатурата. По принцип в него не се вграждат функции за самостоятелно разпространение.

Подтип на spyware е програма **Adware**, която осигурява преноса на събрана потребителска информация. Най-често информацията се изпраща до определени хостове, които връщат обратно рекламна или друга информация и/или програми до заразената машина.

Вариант, попадащ в категорията на **spyware,** е **tracking cookies**. **Бисквитките** (Cookies) се използват в Internet-базираните системи. Те записват информация за дейността на потребителя (на локалната машина) при влизане и ползване на услугите на даден сайт. Събраната информация за посетените страници от потребителя впоследствие могат да се използват за социален инженеринг.

Заразяването става основно при инсталиране на програми, в които има вграден софтуер за следене. В повечето случаи тази допълнителна функция е описана в EULA[8] (End-User License Agreement) с цел информиране на потребителя. Повечето потребители не четат тази информация, инсталират програмата и по-този начин се заразяват.

Защитата от програми-шпиони се базира на *специализирани програмни средства.* Използването на антивирусни програми и защитни стени *не може* да гарантира необходимата защита.

Един от вариантите за откриване на spyware е *търсенето по предварително известно име на файл*. Предварително известните имена на файлове, съдържащи такъв код, са записани в речник. При откриване на съвпадение на име на файл с файл от речника, първият файл се маркира.

Друг вариант е *анализ на специфични особености на файловете* (file properties). Следи се за размер, име на създател, версия.

Детектирането по файлови атрибути обикновено се комбинира с търсенето по име. По-добро качество на детекция се получава при следене за наличие на предварително дефинирани сигнатури в съдържанието на сканираните файлове. В процеса на търсене на сигнатури може да се търси и наличието на определени команди, които не са специфични за това приложение. (Пример за „странно поведение" е текстов редактор, които притежава команди за отваряне на сокети[9]).

---

[7] Детайлно описание може да се намери в материали [37,38,52,53]

[8] Условията за използване на софтуер от даден производител. Най-често се показват преди да започне инсталацията на продукта.

[9] Уникална комбинация от IP адрес и порт.

Една от важните характеристики на spyware е относително големия размер на кода, с който се разпространяват. Създателите на тези продукти ги скриват на различни места в програми с голям по обем код, с което да затрудняват детектирането на шпионския код.

7.2.6 Капан (задна врата, Trap Doors)

"Задната врата" е метод за ползване на системни ресурси посредством заобикаляне на процедурата по автентификация на потребителя. Тези врати се оставят от дизайнерите (програмистите) на софтуера по различни причини. Най-често това се прави с цел гарантиране на контрол върху действията на техния софтуер, без да може да се следи достъпа им от друго лице. Така се гарантира възможност за неоторизирано проникване в система или програма.

7.2.7 Логически бомби (Logic Bombs)

Логическата бомба е програма или програмен код, изявяващи се спорадично само при определени условия в процеса на работа. Спорадичността е предпочитан начин за изявяване, понеже гарантира незабележимост на логическото условие при верифициране на работата на програмата или софтуера. Елементи на програмния код, които се използват много рядко, е най-вероятно да не бъдат тествани или анализирани. Незабележимостта разчита на тази малка вероятност. Аналогично се допуска, че при тестване на дадена програма няма да се стигне до такава дълбочина на теста, в която да се открие логическа бомба.

Пример: Дадена програма се пуска в експлоатация през 2004 година. В част от софтуерния код, които реализира определена функция се поставя условие, което при активиране на тази функция и констатиране на дата на системния часовник от 2005 година, преустановява работата на програмата и изтрива натрупаните до момента данни. Частта от софтуерния код, реализиращ тази операция, се поставя в рядко използвана функция – например настройването на определена опция. Спорадичното й използване предполага натрупване на голям обем от данни и при евентуалното и активиране (поне след година работа на програмата) се нанасят сериозни щети.

7.3 АТАКИ, НАСОЧЕНИ КЪМ КОМУНИКАЦИОННИЯ СЛОЙ

7.3.1. ARP (MAC)Flooding

ARP е протокол, използван от крайна система за определяне на хардуерен адрес (MAC), съответстващ на търсен IP адрес, с който крайната[10] система иска да комуникира. На това ниво *не съществува* механизъм за автентификация на двете страни в комуникацията.

За да се обясни действието и последствията от тази атака, е необходимо да се припомни разликата между комутируема и некомутируема Ethernet мрежа. Видът на мрежата се определя от мрежовото устройство, което осъществява

[10] Терминът се ползва съгласно RFC 791.

връзката между крайните системи в локалната мрежа. Входната точка за връзка на мрежовото устройство с крайната система (host) се нарича **порт на мрежово устройство**. Когато мрежовото устройство доставя получения на даден порт пакет до всички останали портове (boradcast), то се нарича **хъб** (hub). Ако мрежовото устройство доставя постъпилия на даден порт пакет до точно определен порт, се нарича **суич (switch).** Комутиращите функции на всеки switch се основават на възможност за анализ на пакетите (фреймовете). Всеки фрейм съдържа в заглавната си част MAC адреса на адресираната крайна система. Всеки switch притежава таблица на съответствието, отразяваща кой MAC-адрес на кой порт е включен. В режим на „комутатор" всеки switch използва тази таблица, за да доставя коректно получените пакети.

Когато се използва хъб за реализация на локална Ethernet мрежа, съществуват следните недостатъци:

- поради начина си на действие, хъбът има много колизии[11];
- всяка крайна система получава всеки пакет. Следователно има потенциална възможност, посредством поставяне на мрежовата карта в отворен режим на работа[12] (promiscuous mode), да „подслушва" целия трафик в локалната мрежа.

Програмите за следене на трафика се наричат **снифери** (sniffer) или **протоколни анализатори**. За да получават всички пакети, тези програми поставят мрежовата карта на крайната система в отворен режим и „подслушват трафика", преминаващ през съответния хъб.

При използване на мрежово устройство от тип switch възможността за такова подслушване е ограничена. За да се организира подслушване на трафика, е необходимо да се атакува комутиращото устройство и да се промени неговия режим на работа.

Всяко комутиращо устройство поддържа таблица на съответствие (Content Addressable Memory), която показва кой MAC адрес, към кой порт е свързан. В момента на препълване на тази таблица, комутиращото устройство има нужда да я обнови. За целта то преминава в режим на хъб. В този режим се прекратява комутирането на постъпващите пакети и те се изпращат до всички портове. По този начин те могат да бъдат подслушани.

7.3.2. MAC spoofing

Трафикът, преминаващ през мрежата, може да се подслушва посредством **фалшифициране на физическия адрес** на мрежовата карта. Мрежовите карти се произвеждат с MAC-адрес, който е установен от производителя и се счита за уникален. MAC-адресът представлява 6 байтово число. Първите три байта са уникален код на картата за съответния потребител.

---

[11] Колизията е събитие, при което две мрежови устройства се опитват да предават едновременно данни в един и същ канал и това води до прекратяване на предаването и в двете устройства.

[12] Да приема всички пакети, независимо от адреса на получателя.

Съществуват методи за подмяна на това число или производство на мрежови карти с адреси, дублиращи съществуващите. След извършване на подмяната, е възможно да се получава трафика предназначен за картата (притежаваща същия номер) на атакуваната крайна система. Подробното използване на тази възможност е представено в описаните по-долу атаки.

7.3.3 ARP spoofing

Всяка крайна система, използваща технологиите Ethernet/IP, притежава два адреса:

- хардуерен адрес на мрежовия интерфейс – MAC адрес и
- софтуерно зададен IP адрес, които се въвежда от потребителя на системата (статичен адрес) или/и се присвоява автоматично от друго мрежово устройство (динамичен адрес) в момента на включването й в мрежата.

Когато крайната система изпраща данни към друга крайна система, тя е длъжна да формира пакет (фрейм), в който се записва:

- собствения MAC-адрес на изпращащия фрейма;
- собствения IP адрес на изпращащия фрейма;
- MAC-адреса на получателя на фрейма;
- IP адреса на получателя на фрейма.

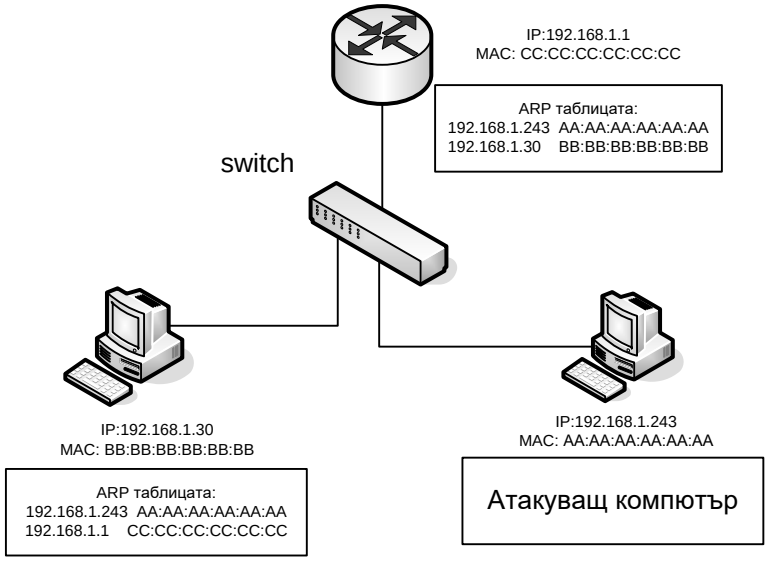
Тези данни са подредени във фрейма съгласно протоколите от различните нива на OSI модела, които се формирали този фрейм. За да се формира фрейма, е необходимо да са известни съответствията между IP адресите и MAC-адресите на двете крайни системи. В момента на конструиране на Ethernet фрейма е известен IP адреса на получателя, но не е известен неговия MAC-адрес. За да разбере този адрес, Ethernet протокола използва ARP (Address Resolution Protocol) протокол. ARP изпраща запитване под формата на пакет (ARP request), с който търси съответствие между IP адреса на получателя на пакета и неговия MAC-адрес. Пакетът се изпраща до всички компютри в локалната мрежа (broadcast). Преведено на литературен език запитването има следния смисъл: «*Този, който притежава търсения от мен IP адрес, моля да ми изпрати като отговор стойността на хардуерния си адрес (MAC адреса)»[45]*.

След като някой хост разпознае в запитването собствения си IP адрес, той изпраща към питащия компютър ARP пакет (ARP replay), в който записва стойността на своя MAC адрес. След получаването на този пакет първият хост може да изготви Ethernet фреймовете и да ги изпрати към получателя им (отговорилия компютър). За да минимизира ARP запитванията, операционната система поддържа локален списък на съответствията между IP адреси и MAC-адреси. Този списък се нарича ARP таблица (кеш, cache).

ARP spoofing е *атака свързана с подмяна на съдържанието на ARP таблицата, целящо фалшифициране на съответствието на IP адрес спрямо MAC адрес.*

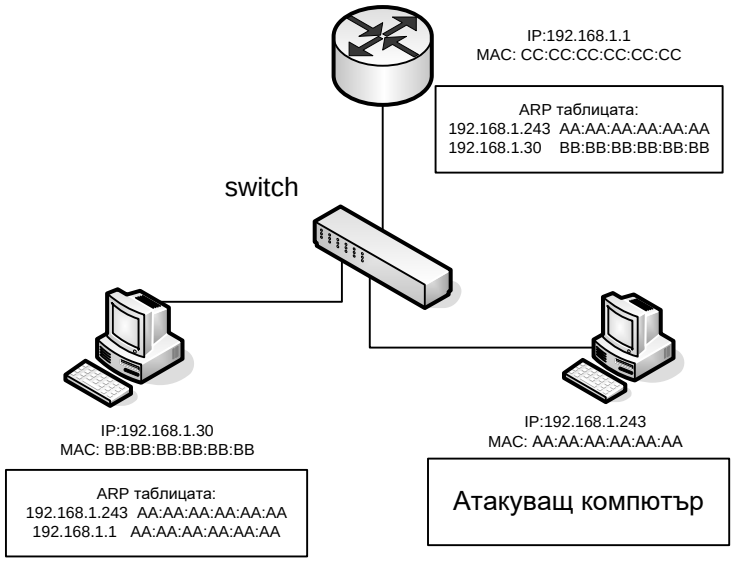
След провеждане на успешна атака, всички пакети изпращани към даден IP адрес ще се изпращат като пакети към подменения MAC адрес, който е адреса на атакуващия компютър. Пример за такава атака е описан в [45 и 54]:

Началното състояние на мрежата е показано на Фиг.7.2. ARP таблиците са попълнени с коректни стойности и атакуващия компютър с адрес 192.168.1.243 няма достъп до пакетите на компютър с IP адрес 192.168.1.30, когато тези пакети не са предназначение за него.



Фиг. 7.2 Начално състояние на ARP таблиците

Атакуващият компютър иска да прихване всички изпращани пакети от компютър с адрес 192.168.1.30. В случая това са пакетите от 192.168.1.1 - рутера на мрежата. При нормалното функциониране на мрежата това не може да стане. Причината е използването на комутиращо устройство (switch), което адресира пакетите на базата на собствената си маршрутизираща таблица. В нея е записано съответствието «MAC-адрес – порт на комутатора».



Фиг. 7.3. Състояние в края на атаката.

Целта на атаката е 192.168.1.30 да изпраща пакети с IP адрес на получателя 192.168.1.1, но с MAC-адрес AA:AA:AA:AA:AA:AA. За да успее тази атака е необходимо да подмени съдържанието на ARP таблицата на 192.168.1.30. Подмяната на съдържанието се извършва посредством изпращане на фалшиви ARP съобщения под формата на broadcast или директно насочени (под формата на ARP replay) към 192.168.1.30. Най-вероятно е атакувания компютър е да е с включена опция за автоматично обновяване на собствената си ARP таблица. Ако това е така, то той приема ARP заявката за обновяване и извършва подмяната на истинския MAC адрес на 192.168.1.1 е MAC адреса на 192.168.1.243.

В края на атаката състоянието на ARP таблицата на атакувания компютър (192.168.1.30) е представено на Фиг.7.3:

Детектирането и противодействието на такава атака е трудно. Най-сигурният вариант е забраната за автоматично обновяване на ARP таблиците на компютрите в локалната мрежа. Допълнителни възможности са:

- конфигуриране на комутиращите устройства да работят в „защитен режим”. За целта те следят за уникалността (единствеността) на възможните комбинации „PORT-MAC” и не обслужват пакети, които не отговарят на това правило;
- следене за наличие на повече от едно съответствие от тип „IP-MAC” и блокиране на трафика към и от крайните системи, за които е регистрирано повече от едно съответствие от системите за детектиране на интрузия[13].

[13] Представени са в глава 9.

7.3.4. Спуфинг на IP адрес (IP Address Spoofing)

Всяко крайно или мрежово устройство притежава уникален IP адрес, който му дава възможност за работа в TCP/IP мрежа. Въз основа на този адрес се организира маршрутизирането и филтрирането на пакети. Филтрирането е процес на определяне на правата на една дейтаграма за достъп или излизане от дадена мрежа. Основната задача на филтрирането е да се пропускат за дадена комуникационна мрежа единствено дейтаграмите, предназначени за нея. Подробно описание на правила за филтриране може да се намери в RFC 1918[14].

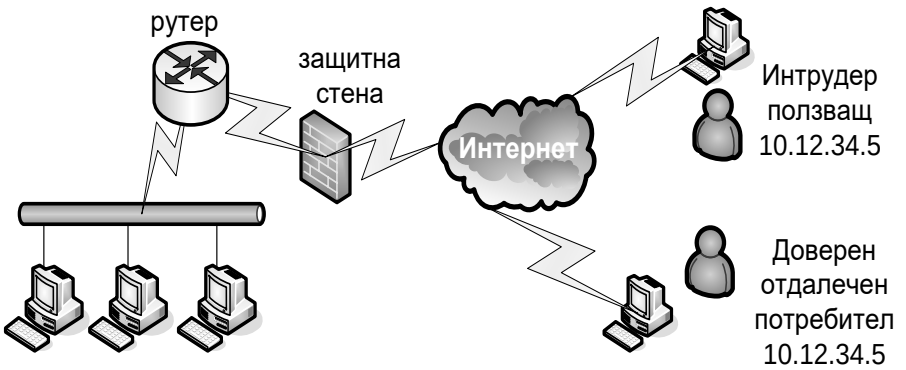
Спуфингът представлява инжектиране на IP дейтаграми в дадена мрежа, чиито адрес на подателя (source address) е фалшив (не е на машината подател, а друг). Целта на спуфинга е атакуващият хост, който се намира извън локалната мрежа, да се представи за доверен хост, който:

- принадлежи на мрежата в която се намира атакувания хост или
- принадлежи на доверено адресно пространство извън рамките на локалната мрежа на атакуваната машина.

За да се проведе тази атака е необходимо да се променят рутиращите таблици на мрежовите устройства по маршрута от интрудера до атакувания хост. В противен случай, дейтаграмите ще бъдат отхвърлени при преминаването им през мрежата.

При успешна подмяна на адресите, отговорите на дейтаграмите, изпратени от атакуваната система, ще бъдат насочени към използвания от интрудера IP адрес. Този факт може да се използва в следните аспекти:

- интрудерът може да получава изпратените данни от атакуваната система и да ги използва;



Фиг. 7.4 Схема на IP spoofing

- отговорите на аткуваната система са дейтаграми, представляващи излишен трафик за мрежата, към която са насочени. Така атакуваната

[14] Request For Comment – отворени стандарти за разработване на мрежови технологии и приложения.

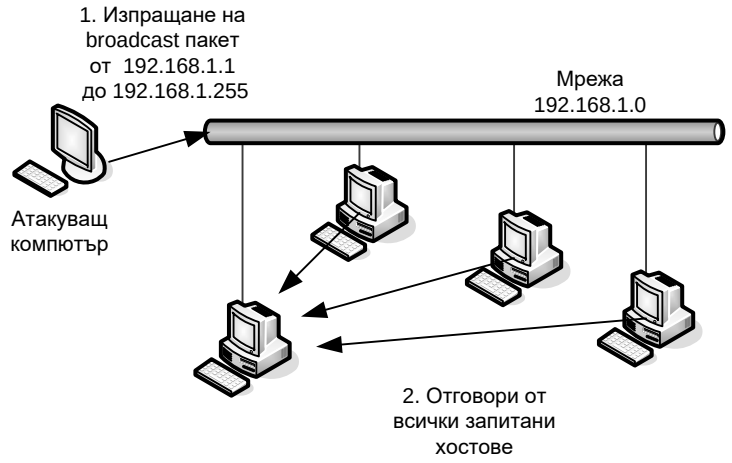
хост-система започва да играе роля на атакуваща система по отношение на третата мрежа, към която тя изпраща излишен трафик;

- интрудерът, след като е инжектирал дейтаграми с фалшив адрес, има възможност да наблюдава отговорите (дейтаграмите) на системите, които е атакувал. По този начин, освен генерирането на излишен трафик (което е вид активна атака), се провежда и втора, пасивна атака, която има за цел да установи принципи на действие на протоколите за управление на сесия (протоколите от транспортния слой). Изследването на техните характеристики дава възможност при следващи атаки, насочени към транспортния слой, да се предскаже поведението на конкретната реализация на даден протокол и да се използва в процеса на интрузия.

7.3.5. Атака за увеличаване на трафика

Атаката с подмяна на адреса на изпращащата страна (IP spoofing) в дейтаграмата може да се използва за увеличаване на трафика в дадена мрежа. Начинът за реализиране на атаката е даден на фиг.7.5.

Дадена е мрежа, обхващаща адресното пространство от 192.168.1.0 до 192.168.1.255. Мрежата е от IP клас C. В нея могат да се адресират 254 хоста. Адрес 192.168.1.0 е адрес на мрежата и не може да се присвоява на отделен хост, а адрес 192.168.1.255 се използва за едновременно адресиране на всички хостове (broadcast). Когато се изпрати пакет до 192.168.1.255, той се изпраща до всички хостове в тази мрежа. В зависимост от съдържанието на пакета хостовете могат да отговорят или не на този пакет. Дали да се отговори или не, се определя от протокола от мрежово ниво. Ако той е TCP, няма да има отговор, понеже TCP е проектиран да изгражда връзка между два хоста и не може да работи с broadcast адрес. Ако обаче протокола е UDP или ICMP, може да се очаква обратен отговор.



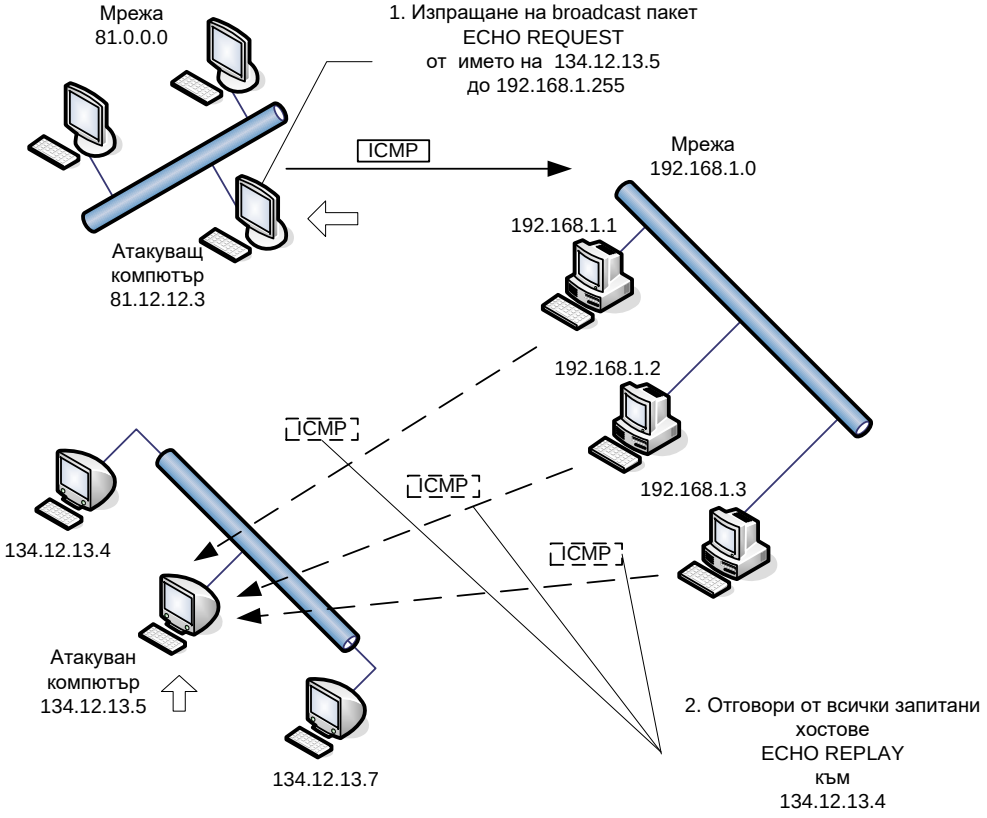
Фиг. 7. 5 Атака за увеличаване на трафика

7.3.6 Smurf Attack

Името на тази атака произлиза от името на сорс кода, с които е реализирана за първи път. (smurf.c).

Атаката се базира на изпращане на дейтаграма с подменен IP адрес до хостовете в дадена мрежа. Адресът на получателя (Destination Address) на дейтаграмата е подменен – в него е записан адреса на атакувания компютър.

Вариант за провеждане на такава атака е представен на фиг. 7.6. Атакуващият компютър е с адрес 81.12.12.3. Изпратената дейтаграма от атакуващия компютър е от тип ECHO REQUEST на ICMP протокол. В тази дейтаграма е подменен адреса (Source Address) на подаващата страна. Вместо да запише своя адрес (81.12.12.3), атакуващия хост изготвя дейтаграма, в която (в полето SA) записва адрес на компютър от мрежа 134.12.13.0. В случая е записан адрес на хост 134.12.13.5. Дейтаграмата се изпраща до всички компютри в мрежа 192.168.1.0 посредством IP network broadcast addresses – в случая 192.168.1.255. Всички адресирани компютри отговарят с ICMP: ECHO_REPLAY, като адресът, на който трябва да се върне отговора, е 81.12.13.5. Така при връщане на отговора се "наводнява" мрежа 134.12.13.0 с голямо количество ненужни пакети.



Фиг. 7. 6 Провеждане на smurf атака

7.4.6. Атаки, базирани на промяна на размера на дейтаграмата

IP заглавната част (header) притежава 16 битово поле, за записване на размера на дейтаграмата. Максималната дължина на един пакет е 65 535 байта. Дължината на заглавната част на дейтаграмата се записва в поле IHL. Тя се измерва в брой на 32 битови думи, съставящи заглавната част на дейтаграмата. При преминаване на един пакет през мрежа, която има размер на фрейма, помалък от текущата дължина на пакета, дейтаграмата се фрагментира. Разделя се на две или повече дейтаграми, чийто размер позволява преминаване през конкретната мрежа. Този процес се нарича асемблиране на дейтаграма. При приемането на дейтаграмите е необходимо да се реасемблират. В процеса на реасамблиране на дейтаграмата възникват следните проблеми:

- При пристигане на фрагмент, който е дубликат на пристигнал порано друг, как да се разбере кой е истинският?
- Ако пристигне фрагмент, припокриващ частично съдържанието, което е реасемблирано до момента, трябва ли този фрагмент да се отхвърли или да се приеме?
- Как да се обработи дейтаграма, в която има несъответствие между размера, записан в IHL, и размера на цялата дейтаграма?

Възможна атака, основана на превишаване на максимално допустимата дължина на пакета, е **Ping Of Death.** Тази атака [45,54,39] разчита на грешка в реализирането на софтуера за работа на ICMP протокола. Максималният офсет на полезните данни в една дейтаграма може да е 65 507. Ако този офсет на фрагмента е записан в една дейтаграма, това означава, че полезния товар (данните) не могат да бъдат повече от 7 байта. (Пресмята се като разлика от тоталната дължина на дейтаграмата минус стойността записана в IHL). При използване на локална мрежа с Ethernet-технология (максималната дължина на фрейма може да е 1 500 байта), интрудерът може да създаде пакет с офсет 65528 и данни с размер 1480 байта, което при реконструиране дава пакет с дължина 67 008 байта. Ако този пакет се приеме от софтуера, реализиращ IP протокола, има възможност при заделянето на памет да се получи препълване на буфера (превишаване на разрешената за заемана памет от програмата). Това може да се отрази на останалите програми и да промени данните, с които работят. В този случай в операционната система възниква отказ, който може да доведе до нарушаване на функционалните алгоритми и излизане от работоспособно състояние („забиване").

7.3.7. Атаки, насочени към подмяна на съдържанието на заглавната част (header) на ICMP протокола

Подмяната на ICMP (Internet Control Message Protocol) пакети има най-често за цел да извърши атака от тип „отказ от услуга" (denial-of-service) посредством фалшива сигнализация за недостъпна мрежа или хост. ICMP е протокол, който не установява сесия и е невъзможно да се въведе автентификация на постъпващите ICMP пакети. Реализирането на такава атака се свежда до изпращане на пакети до атакувания хост, съдържащи съобщение от типа на „**Destination unreachable**"[15].

[15] Подробно описание на действието на ICMP може да се намери в RFC 792 и [53]

Изпращането на пакет с този формат на ICMP отговора представлява фалшиво съобщение за невъзможност за достигане до дадена мрежа или хост. Ако атакуваната крайна система е сървър, предоставящ услуга, това е един от начините да се блокира достъпа до него без да се атакува самия хост. За целта е необходимо да се атакува крайната система, която има роля на gateway, за мрежата и от нейно име да се изпращат такива пакети.

Друг вариант е изпращане на съобщение: „**Host Quench**” .

Изпращането на такъв пакет се прави в случай, че дадена крайна система не може да приема в момента пакети от изпращащата страна поради препълване на входния буфер. При получаване на това съобщение изпращащия хост е длъжен да намали скоростта на предаване на данни.

7.3.8 Лошо съчетаване на флагове в заглавната част на TCP протокола

Тази атака се основава на изпращане към атакуваната система на невалидни съчетания на флаговете на TCP хедъра. Различните приложения на TCP-стек реагират по различен начин на такива комбинации. Някои стекове губят работоспособността си, с което се отнема възможността на операционната система за мрежови комуникации. В *Таблица 7.1* са показани част от невалидните комбинации, които могат да се използват за атака.

Таблица 7.1

Комбинация от флагове	**Възможен проблем**
Няма установен флаг	Този пакет не може да се класифицира и има вероятност протоколния стек да го обработи неправилно или да загуби работоспособността си.
SYN = 1 FIN = 1	Тази комбинация индицира едновременно инициализиране на сесия и завършване на сесия.
SYN = 1 RST = 1	Тази комбинация индицира едновременно стартиране на сесия и прекъсване на сесия.
SYN = 1 ACK = 1 FIN = 1	Комбинацията на тези флагове показва едновременно отговор на инициализиране на сесия и прекъсване на сесия.
SYN = 1 RST=1 FIN = 1	Комбинацията на тези флагове показва инициализиране на сесия, прекъсване на сесия и приключване на сесия.
Установяване на всички флагове в 1.	Тази атака се нарича Xmas Tree флагова комбинация. Тя представлява едновременно желание за начало на сесия, отговор и прекъсване.

Всяка една от тези невалидни комбинации може да даде различни откази в конкретната реализация на протоколните стекове, която да влияе по различен начин на работоспособността на комуникационните системи.

7.3.9 Сканиране на портове (Port Scanning)

Всеки интрудер има за цел да придобие информация за системата, която ще атакува. За целта той иска да разбере типа на операционната система, програмите, които работят в нея, какви комуникационни ресурси са достъпни за потребителите й, естествено, начина им на защита. Обикновено комуникационните ресурси се активират като клиенти или сървъри. За да могат да работят, те се нуждаят от портове за да образуват сокет. Портовете представляват определени адреси в оперативната памет. В тези адреси се записват постъпващите пакети за приложенията, които трябва да ги ползват. Всеки порт има номер. Номерата на портовете са от 0 до 65535. Портовете с номера от 0 до 1023 са резервирани подразбиране за услуги, които се ползват в комуникационните мрежи. Една такава услуга е Hyper Text Transfer Protocol. Сървърите, предоставящи тази услуга, заемат порт 80 в момента на своето стартиране, където очакват получаване на пакети от програмите клиенти.

Програмите за сканиране на портове имат за цел да търсят такива отворени портове. При установяването на активен порт се анализира софтуера, работещ чрез него, с цел да се атакуват слаби места в този софтуер и чрез него хоста или мрежовото устройство, на което е стартиран.

Противодействието срещу сканирането се реализира от системи за детектиране на интрузия чрез извършване на обратно проследяване. Това е сложен процес, понеже е трудно да се прецени, кое сканиране е интрузия и кое е начало на сесия. По-голямата част от програмите-скенери търсят отворени портове посредством изпращане на пакети за откриване на сесия - SYN=1 (TCP). След изпращането на първия пакет се установява полуотворена връзка (RFC 793), която трудно се класифицира като интрузия.

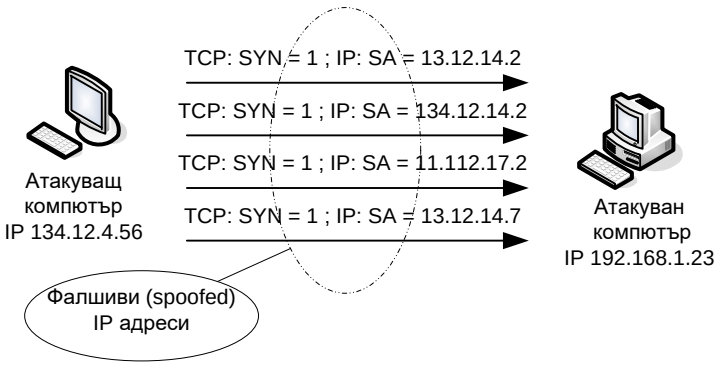
7.3.10 SYN flooding

Някои от реализациите на TCP стека допускат провеждане на атаки, чиито резултат е лавинообразно отваряне на портове - имитация за начало на връзка. Провеждането на тази атака има за цел да заеме максимален брой портове с фиктивни връзки. По този начин се блокира възможността на операционната система да обслужва постъпващи заявки и става недостъпна за отдалечени потребители на стартирани услуги в нея.

Атаката се състои в изпращане на TCP пакет с установен флаг SYN = 1. Атакуваната система отговаря с TCP пакет с установени SYN = 1 и ACK = 1. При отговора се създава TCB (transmission control block), в който се записва текущото състояние на сесията и се преминава в състояние „чакане на потвърждение" от започналия сесията (съгласно three-way hand shake процедурата за договаряне на сесия). Това състояние на сесията се нарича режим на полуотворена сесия (*half-open connection).*

Ако всички тези SYN пакети са с невалиден адрес на източника (IP spoofing), няма да бъде получен очаквания отговор за приключване на процедурата по договаряне на сесия. «Времето за живот» на всяка една от полуотворените сесии е зададено и, след неговото изтичане, връзката се унищожава. Но в рамките на това време тя е заела ресурс (порт, памет), който не може да се използва за друг процес. Увеличаването на тези съединения може да изчерпи ресурса, който е заделен за тях, при което се блокират комуникационните ресурси на съответния хост. Това е и всъщност целта на атаката.

Състоянието на хост по време на провеждането SYN flooding е показано на Фиг.7.7



Фиг. 7. 7 SYN flooding атака

Възможни са следните стратегии за противопоставяне на тези атаки:

- лимитиране на броя отворени съединения (конекции) за определен интервал от време.;
- прекратяване на част от конекциите и подмяната им с постъпили нови;
- динамично пренастройване на правилата за филтриране на пакети с цел недопускане на пакетите на интрудера;
- използване на SYN cookies.

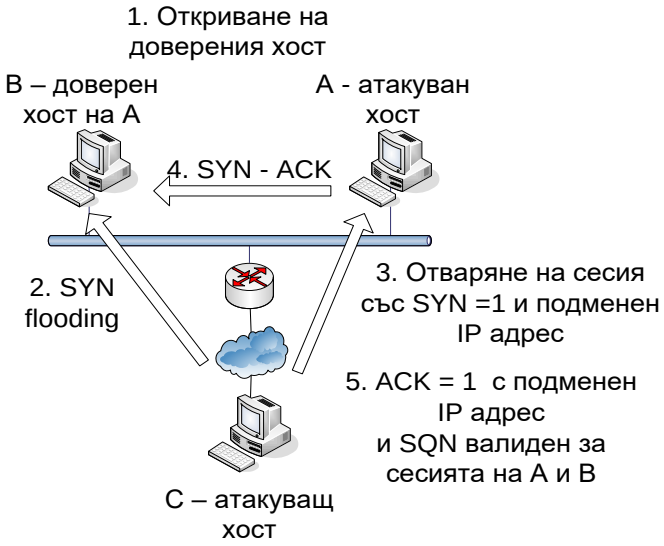
В SYN cookies-техниката поредният номер на сегмента е защитен криптографски. При изпращане на SYN пакет хостът, който го получава, отговаря с ACK/SYN пакет, в който се изпраща криптиран[16] пореден номер. При този отговор не се отваря полу-отворена връзка. Счита се, че получателят на дейтаграмата ще успее да декриптира номера и да отговори с правилния пореден номер. Едва тогава се започва изпълнението на стандартния three-way hand shake алгоритъм за договаряне на сесия.

7.4.11 Спуфинг на поредния номер (Sequence Number Spoofing)

[16] Криптографията е обяснена в глава 10

TCP/IP протоколният стек номерира уникално (Initial Sequence Number) всеки пакет, изпратен от конкретен сокет в процеса на предаване (RFC 793). Генерирането на номер за първия пакет се извършва от всяка операционна система. За генерацията се използва системния часовник и функция, работеща с него – генератор на случайни числа. Откриването на правилата за работа на алгоритъма, генериращ номера, води до възможност за отгатване на поредните номера. На този основа е възможно произвеждането на фалшиви пакети, с които да отговори при установяване на връзката. При успешен отговор интрудерът заблуждава другата страна в комуникацията.

Реализирането на такава атака, съвместно с подмяна на адреса на получателя на пакета, дава възможност за получаване на контрол върху атакувания компютър. Демонстрация на начина на провеждане на атаката [39,45,46] е дадена на Фиг. 7.8.



Фиг. 7.8. Спуфинг на поредния номер

Целта на атаката на C е да получи достъп до компютъра на A, да инсталира свой софтуер, чрез който впоследствие да управлява компютъра на A.

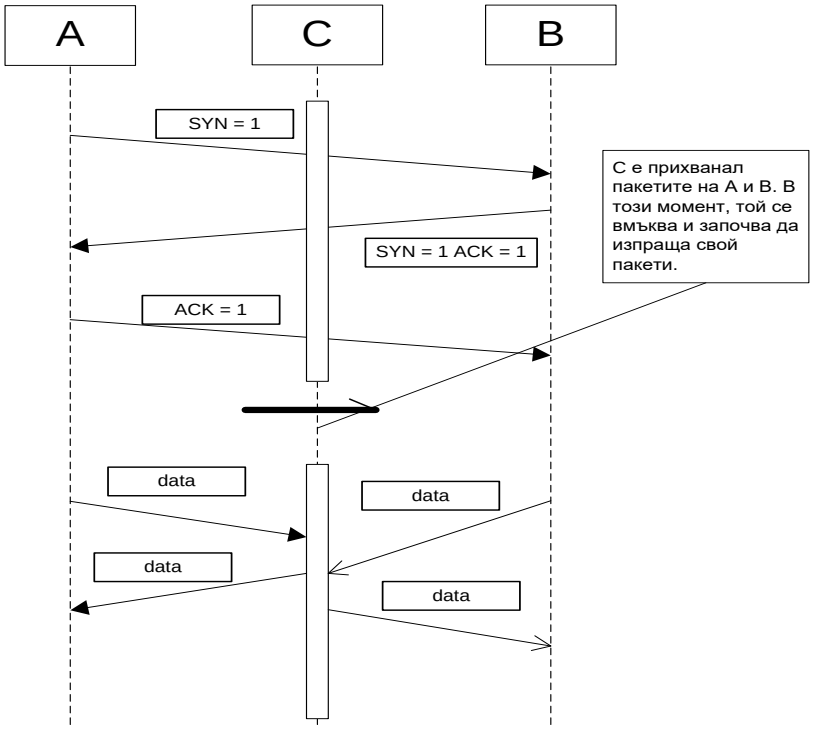
Атаката се реализира на пет етапа:

1. **C** трябва да открие кои са хостовете, на които **A** има доверие, тоест им позволява да изпълняват команди и файлови операции на собствения си компютър. Такъв доверен хост в случая е **B**.
2. **C** започва атаката си като първоначално се опитва да изолира компютъра на **B**. Изолирането на **B** се извършва като му се отнема възможността за мрежова комуникация. Това става посредством SYN flooding.

3. **След** като **B** е изолиран, C започва процедура по отваряне на сесия с **A.** Изпраща се TCP пакет с флаг SYN = 1. В този пакет **C** записва IP адреса на **B (**IP spoofing**).**
4. **A получава** пакета и по IP адреса решава, че дошлия пакет е от **B**. **A** отговаря на получения пакет, като за целта изпраща към **B** TCP пакет, в който флаговете ACK и SYN са установени в 1. В случая **B** не може да върне към **A** TCP пакет с установен флаг RST =1, за да прекъсне сесията, защото комуникационните му възможности са блокирани от атаката, проведена на етап 2.
5. На етап 5 **C** изпраща TCP пакет към **A,** в който поредния номер е подменен с валиден (от гледна точка на B) и флага ACK е установен в 1. **C** атакува и рутера на мрежата на B и подменя в него маршрутизиращата таблица така, че изпращаните към **B** пакети да се отклоняват към **C.** След получаването на този пакет сесията между **A** и **C** е изградена и преминава в състояние ESTABLISHED. В това състояние A може да инсталира на компютъра на B своя програма, която да използва в последствие или да промени настройките по сигурността на компютъра на **A**

7.3.12 Session Highjacking

Тази атака е вариант на TCP/IP spoofing. Целта на атаката е вмъкване в сесията между клиент и сървър. При успешното вмъкване интрудерът получава изпратените от всяка страна пакети и ги подменя със свои. Схемата на атаката е дадена на Фиг.7.9. **C** се вмъква в сесията между **A** и **B** след като тя премине в режим ESTABLISHED. При вмъкването, когато пристига пакет за **B**, **C** му отговаря, като вместо своя IP адрес, изпраща обратно пакет с IP адреса на **B.** По същия начин **C** се представя за **A,** когато трябва да отговаря на пакетите на **B**.



Фиг. 7.9. Session Highjacking

Провеждането на тази атака се основава на придобиването на неоторизиран достъп до рутер или аналогично устройство, работещо като шлюз (gateway) между легитимния потребител и сървъра, с който той работи в момента. Неоторизираният достъп до мрежовото устройство и поемане на управлението му се извършва посредством придобиването на права за администриране.

7.4.14. DNS

Domain Name Service (DNS) е йерархична система, използвана за осигуряване на работата на TCP/IP протоколния стек. Основната й функция е да осигури транслирането на имената на domain-ите в IP адреси. Имената на domain-ите се използват в URL адресите.

Пример: При търсене на достъп до HTML сървър обикновено потребителят се обръща към него по име. Примерно, изписва в полето за URL адреса: http://www.tu-sofia.bg. Това име не може да се използва от IP протокола. Необходимо е да се намери кой IP адрес съответства на името. След намирането на IP адреса, към него започва и изпращането на пакетите, съдържащи потребителската заявка. За да се определи IP адреса, се използва услугата DNS. В IP базираните комуникационни мрежи се

инсталират DNS сървъри. В тези сървъри се разполагат бази данни, в които са записани съответствията между IP адреси и имена на хостове. Когато се използва DNS услугата, всеки хост изпраща запитване към известен на него DNS сървър. Последният му отговаря, като в отговора записва търсени от хоста IP адрес. В случая за www.tu-sofia.bg това е IP адрес 81.161.241.11.

Към софтуера, отговарящ за това преобразуване, са насочени и голяма част от атаките. Целта на тези атаки е подмяна на DNS таблиците на съответствие. При това се подава фалшив IP адрес, когато се търси дадено име. Хостът, получил този IP адрес, ще се обръща към него за търсене конкретна услуга.

Пример: Интрудерът успява да регистрира име www.download.com в определен DNS сървър, като IP адресът свързан с това име, не е истинският адрес, а адрес на предварително подготвен от интрудера хост. Всяко търсене на услуга от оригиналния сайт се пренасочва към IP адреса, върнат от DNS сървъра – тоест към хоста на интрудера. Ако интрудера е разработил сайт, сходен с този на търсения от потребителите, то има възможност да ги заблуди и да предостави възможност за download на софтуер, в които има вградени различи програми и средства за придобиване на контрол върху техните машини.

Противодействието на тази атака е невъзможно. Няма обективен начин за доказване на пренасочването. Единственият възможен начин за предотвратяване на атаката е при използване на SSL слоя за криптирано предаване на информация. В този случай се разчита на малката вероятност за притежаване на валиден цифров сертификат от страна на хакера.

7.3.15 Снифинг (Sniffing)

Мрежовият снифинг (снифинг на пакети) е процес на мониторинг на мрежата и събиране на информация за мрежовите ресурси, стартирани процеси на отделните машини и отворени портове. Снифингът се извършва като програмата за снифинг конфигурира мрежовата карта (NIC) да работи в режим на пропускане на всички пакети (promiscuous mode). Всеки кадър в мрежата е адресиран или до всички машини или до машина с определен (уникален) MAC-адрес. При използване на режим на пълно пропускане всички пакети се изпращат към протоколния стек и са достъпни за анализ.

В мрежовия трафик част от мрежовите приложения пренасят паролите в открит вид. Най-често това се извършва при пренос на данни чрез http, ftp, rlogin или telnet. Прихващането на тези пароли е една от целите на подслушването. Другата цел е събиране на информация за различните мрежови ресурси. Въз основа на тази информация се прави анализ за откриване на слаби места в реализацията им и проектиране на възможни атаки.

Най-доброто противодействие е скриване на специализираните мрежови устройства или ресурси, както и задължително криптиране на паролите преди предаването им по комуникационния канал.

Друг вариант за противодействие е откриването на неоторизираните снифери в мрежата. За целта се използва специализиран софтуер, наречен **анти-снифер**. Анти сниферите сканират за наличие на мрежови карти, работещи в

promiscuous mode режим. При откриване на такава карта се прекратява изпращането на пакети към нея.

Защита срещу снифери, до определена степен, позволяват и мрежовите комуникационни устройства, които комутират Ethernet пакети. Използването на суичове (switches) вместо хъбове (hubs) е вариант на защита, понеже доставката се извършва на конкретен порт, при което всяка мрежова карта получава само трафика, предназначен за нея.

7.3.16 Отказ на услуга (Denial of Service, DoS)

Отказ от услуга е интрузия, която има за цел да прекрати или блокира услуга или работата на дадена крайна система като системата се приведе в неработоспособно състояние. Неработоспособното състояние се получава при изчерпване на изчислителните ресурси (процесорно време или памет) или изчерпване на свободните портовете на операционната система. Следователно, реализацията на атаката изисква:

- разпространение на атакуващата програма на голям брой хостове;
- едновременно стартиране на атаката от заразените хостове към крайната цел;
- поддържане на атаката чрез непрекъснато заразяване на нови хостове с атакуващата програма и стартиране на атака от тях.

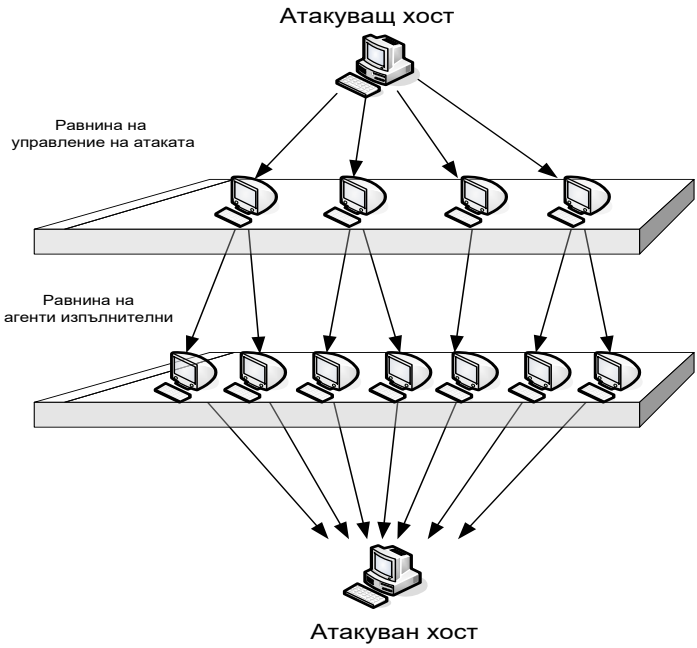
При започване на атаката е голяма вероятността тя да бъде детектирана и хостовете – източници на атаката да бъдат изолирани. Това би довело до прекратяване на атаката. За да се поддържа атаката е необходимо да се разделят атакуващите хостове на два типа:

- хостове, които разпространяват програмата за атака, но не атакуват крайната цел, тоест остават скрити (управляващи агенти) и
- хостове, които реализират атаката към крайната цел (агенти изпълнители).

Управляващите агенти образуват равнина на управлението на атаката, а изпълнителните агенти – равнина на изпълнение на атаката. Отказът от услуга може да се предизвика на различни нива в комуникационния стек. За целта се използват протоколите ICMP, IDP, TCP.

Възможни са два начина за реализиране на тази атака:

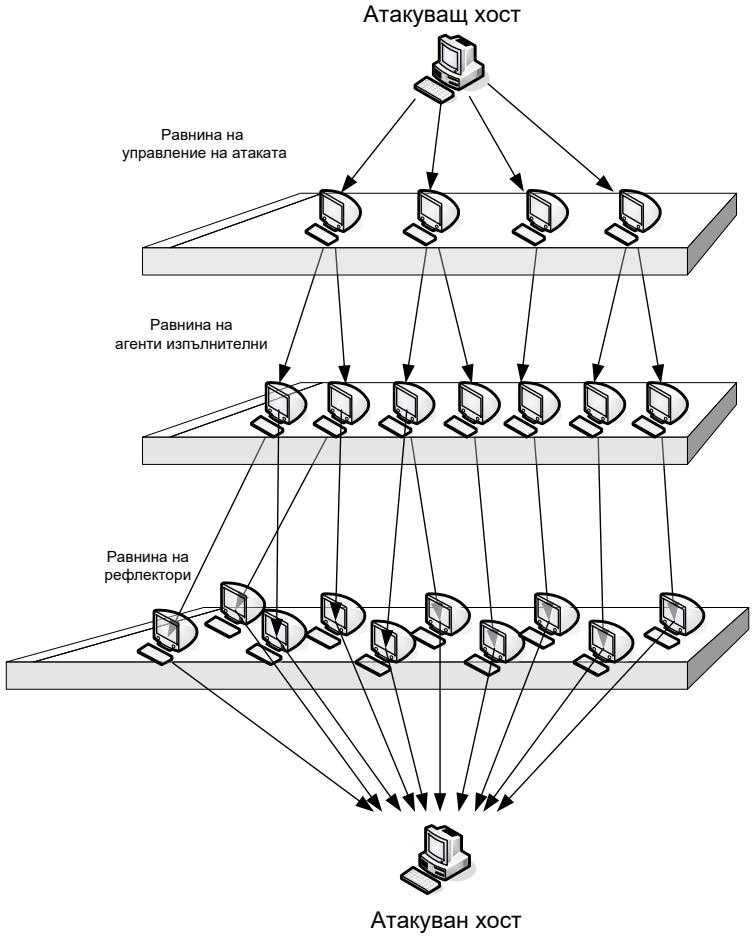
- директен DoS (direct attack);
- DoS с използване на посредник (reflector attack).



Фиг. 7. 9 Схема на директна атака

На фиг.7.9 е представена схемата за реализиране на директна атака DoS [45,46]. При нея агентите-изпълнители директно атакуват ресурсите - цел на атаката и се опитват да ги задържат за себе си или да ги разрушат. На фиг. 7.9 представената атака е изпълнена посредством използване на **хостове-рефлектори.** Атаката се провежда аналогично на предишната, но в случая обекти на атаката са рефлекторите. Рефлекторите отразяват атаката, но отразената от тях се явява нова атака по отношение на атакувания хост.

Пример: При използване на ICMP, агентите изпълнители изпращат ECHO REQUEST към рефлекторите. Пакетите на тази заявка са с подменен адрес на изпращащата страна. В полето SA е записан IP адреса на атакувания хост. В този случай, рефлекторите връщат ECHO REPLAY към атакувания хост, а не към равнината на агентите изпълнители.



Фиг. 7. 2 Схема на атака с използване на рефлектори.

ЗАДАЧИ ЗА САМОСТОЯТЕЛНА РАБОТА

1. Предложете вариант за защита от ARP Spoofing.
2. Предложете вариант за защита IP Spoofing.
3. Предложете алгоритъм за защита от атаки в транспортния слой.
4. Предложете начин за защита от сканиране на портовете.
5. Какви са възможностите за защита от разпределена атака?
6. Как бихте защитили дадена мрежа от SYN Flooding атака?
7. Намерете програми за защита от снифери и обяснете принципите им на действие.
8. Как бихте се предпазили от DNS атака?