Replay напад и методи за намалување на ефектите на овој напад кај бежичните мрежи

Магдалена Тренчевска[1], Александар Јовановски[2], Митко Богданоски [3]

[1,2] Европски Универзитет – Скопје, Р. Македонија,
{magdalena.trencevska,jovanovski.aleksandar}@live.eurm.edu.mk

[3] Воена академија – Скопје, Р. Македонија, mitko.bogdanoski@eurm.edu.mk

***Абстракт*— Replay нападите се многу чести напади и нивното спречување е прилично скапо и проблематично. Постојат повеќе видови на безбедносни мерки кај овој напад. Во овој труд, најпрво ќе биде објаснет replay нападот, која е неговата улога и намера и како настанува тој. Понатаму ќе бидат опфатени повеќе методи за спречување на ефектите од replay нападот. Ќе бидат обајснети неколку методи за превенција, меѓу кои и методот за поставување на знаци на сесија, метод за поставување временско обележје и методот со употреба на т.н Bloom филтер. На крај е направена споредба помеѓу овие методи, со истакнување на најдобрата контрамерка од разгледуваните опции.**

***Клучни зборови*— безбедност, време, знак, методи, replay напад, сесија, филтер.**

I. Вовед

Replay нападот претставува напад на преносот на податоци, каде што како резултат на него пакетот на податоци е злонамерен, лажно се повторува или пак се одложува [1]. Овој напад се изведува на тој начин што злонамерниот напаѓач ги пресретнува податоците и ги ретрансмитира. Како најчест одговор за тоа зашто настанува овој напад се насочуваме кон недостатокот на дизајн на протоколите. Тоа е една од основните причини за случување на replay нападот во бежичната мрежа при транспорт на податоци. Доколку нема посебни предупредувања, контроли и проверки, многу едноставно е да се изврши напад. Replay нападот е мошне едноставен. За успешно реализирање на овој напад потребно е да се искористат заробените пакети, со чие препраќање би се предизвикале неочекувани резултати [1]. Доколку не постои некој начин, односно некој механизам или метод кој ќе врши контрола, односно ќе проверува дали постои некакво дуплирање на пакетите, тогаш replay нападот ќе биде успешен.

Replay нападот всушност претставува нару-шување на безбедноста во мрежата, каде информацијата е складирана без авторизација и потоа пренасочена за да го намами примачот кон неовластени операции, како на пример погрешна идентификација или автентикација или пак дуплирање на трансакцијата. На пример, пораките од еден корисник кој е логиран на некоја мрежа можат да бидат заробени од страна на напаѓачот и препратени наредниот ден. И покрај тоа што пораката може да е енкриптирана, и напаѓачот може да го знае точниот клуч или лозинка, овој напад може да биде усшешен и напаѓачот да добие пристап до мрежата.

Replay нападот, исто така, може да биде успешен во случај комуникацискиот канал да користи криптиран канал кој имплементира силна автентификација. Таков пример е користењето на дигитален потпис. Во тој случај, единственото нешто што напаѓачот нема да го знае е точниот број на трансмисии што треба да се реплицираат, меѓутоа тоа не е од големо значење кога станува збор за бежичните мрежи. Replay нападот за разлика од другите напади слични на него, трае повеќе време. Меѓутоа, неговото ефикасно искористување мора да биде за време на траење на сесијата, односно во време кога се врши транспортот на пакетите од еден корисник на друг или пак од корисник на сервер.

Replay нападите најчесто ги напаѓаат криптираните комуникацисли канали, како на пример IPSec тунелите или пак бежичните протоколи, како и веб апликациите [2]. Постојат повеќе методи за намалување, како и спречување на ефектите на replay нападите. Некои од побитните и најкористените ќе ги разгледаме во овој труд. Врз база на направениот преглед над дел од механизмите, ќе донесеме заклучок за тоа кој метод нуди најсодветно решение против ефектите на овој напад. Во првиот дел ќе биде разгледан методот на поставување на знаци на сесиите (session token). Во вториот дел ќе биде разгедан методот на поставување т.н временски ознаки (timestamps). Како последен механизам против овие напади, кој ќе биде разгледан во овој труд е методот со употреба на т.н Bloom филтер.. На крај, врз основа на направената анализа на претходно споменатите механизми, ќе го препорачаме најдобриот и најсоодветни механизам за справување со replay нападите кај бежичните мрежи.

II. ПРЕГЛЕД НА МЕТОДИ ЗА АНАЛИЗА

A. Поврзани истражувања

Replay нападите постојат долго време. Пред неколку години тие претставувале само напад на лозинка или збор. Денеска тие вршат напад на разни податоци и ги пробиваат без разлика дали имаат дигитални потписи или клучеви. Овие напади од ден на ден сè повеќе и повеќе се развиваат и стануваат многу опасни во транспортот на податоци кај бежичните мрежи.

Според авторите во [2] постојат повеќе техники за спречување на нападите. Според нив најбезбеден начин е да се користат криптографските техники. Овие техники се користат и за спречување на активните и пасивните напади кои често се искористени кај бежичните мрежи.

Во [3] се врши анализа и се дава објаснување на веб услугите и на нивната улога. Во еден дел опишан е преносот на податоци и колку е битна заштитата од нападите. Авторите на трудот тврдат дека како најкористен метод за превенција од напади е идентификацијата на сесијата, која се користи за криптирање на http барањето и повратниот одговор. Протоколот во кој се одвива овој трансфер на податоци се нарекува SSL (Secure Socket Layer) кој користи јавен клуч за размена на клучот на сесијата помеѓу клиентот и серверот.

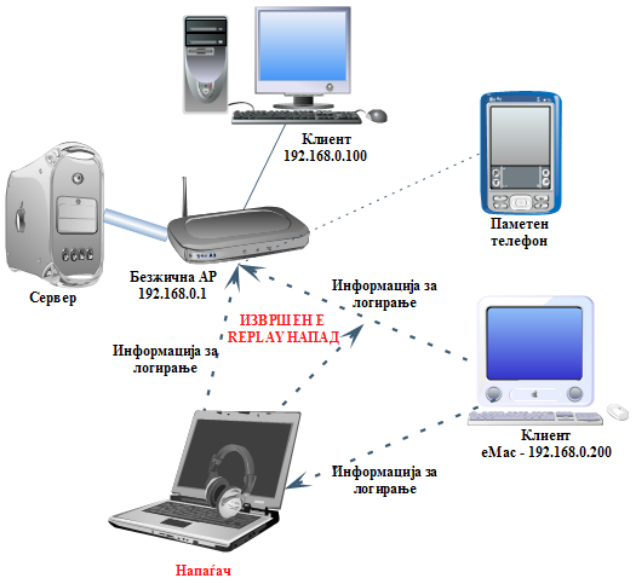
Авторот на [4] во својата книга нагласува дека бројот на идентификацијата на сесиите мора да е уникатен и непредвидлив. Тој вели дека ваквиот знак мора да се прави според случаен избор за да се направи заштита од нападот.

Во статијата [6] е објаснета временската ознака како метод за превенција од replay нападите. Таа е дефинирана како вредност на објект која укажува на времето на системот на критичните точки во целата историја на објектот.

Во [7] авторот вели дека bloom филтерот како метод за превенција од replay нападот е наједноставен. Тој вели дека во споредба со него, останатите методи имаат две негативни страни и тоа: можат да поддржат само лимитиран број на клиенти во случај кога се ограничени и да користат мала количина на меморија и втората негативна страна е дека тие имаат потреба од поголема комплексност на автентикација на клиентите.

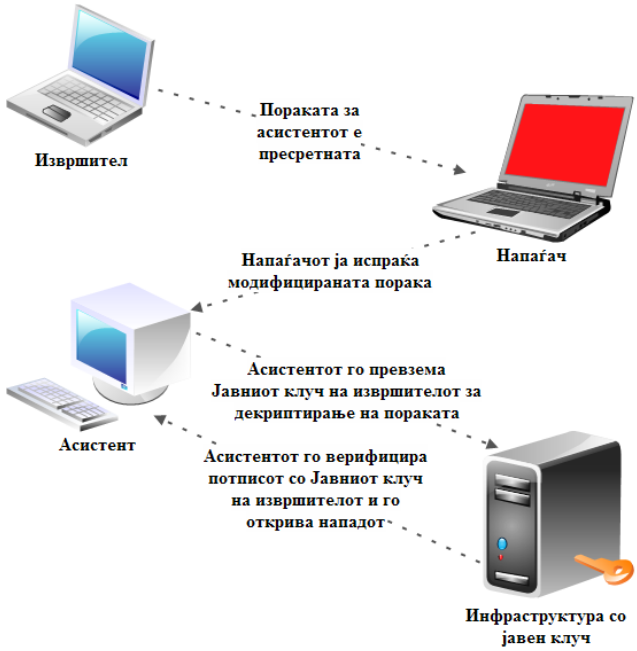
B. Анализа на Replay нападите

Replay нападот претставува напад на податоци за време на нивниот пренос. Тој напаѓа на тој начин што зема дел од податоците и врши промена на истите, односно додава или одзема дел од податоците кои се пренесуваат во мрежата.



Слика 1: Replay напад

На Слика 1 е претставен едноставен начин на replay напад. Клиентот сака, преку пристапната точка, да се логира на серверот. Тој ја праќа својата порака за информација до пристапната точка. Бидејќи пораката не е заштитена истата ја зема и напаѓачот. Тој ја зема кај себе пораката, ја менува и ја враќа користејќи го истиот протокол протоколот преку кој треба да стигне до серверот. Серверот не ја добива пораката во оној облик кој што ја очекува со што клиентот не успева да се логира. За ваквиот тип на напад велиме дека е успешен напад.



Слика 2: Размена на податоци при repaly напад

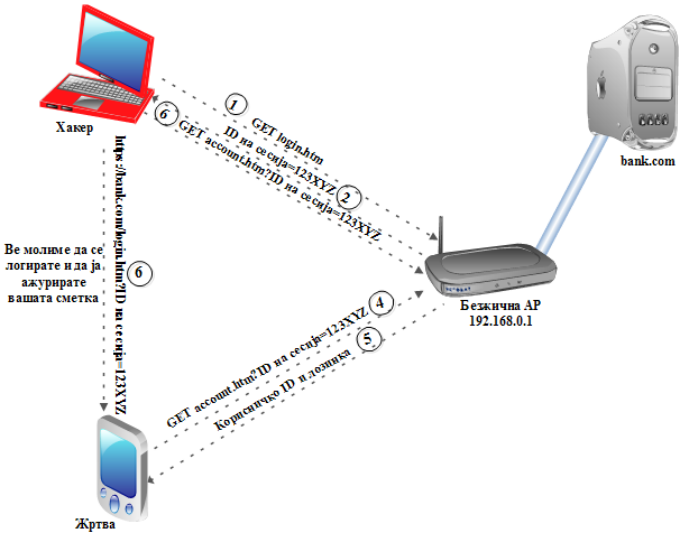
Replay нападот има големо значење и делува негативно во бежичната мрежа. Токму поради тоа, предложени се повеќе методи кои се користат за намалување, како и за целосно спречување на ефектот од replay нападот.

Најчесто користени контрамерки против овој напад се користење на знаци (идентификатори) на сесии, користење на временски ознаки и т.н bloom филтери.

III. Методи за намалување на ефектот на replay нападите

A. *Знак на сесија (Session token)*

Знакот на сесијата (session token) претставува посебен вид на знак (токен) на комукација, познат и како дијалог или конверзација помеѓу два или повеќе комуникациски уреди или пак помеѓу компјутер и корисник во бежична мрежа.[2] Истиот се нарекува и идентификатор или ID на сесија. Овој знак се користи на тој начин што тој се дава на корисникот тогаш кога доаѓа барањето од страна на легитимниот корисник. Тогаш овој знак треба да биде поднесен од корисникот, без разлика дали тој праќа последователно барање, така што серверот ќе биде во можност да го провери овој знак на сесија со знакот кој веќе е складиран во серверот [2]. Овие знаци се бираат преку т.н. случаен процес и тие мора да се менуваат за секоја следна нова акција од страна на корисникот. Ова работи на тој начин што се создава случаен знак секој пат кога сакаме да испратиме некој пакет, односно порака. Знакот го заштитуваме на тој начин што го криптираме и како таков го праќаме на клиентот. Така на пример, секогаш кога клиентот ќе прати потврда дека ги примил податоците се споредува знакот со знакот кој е зачуван на серверот и се гледа дали пораката може да се прифати или не.



Слика 3: Превенција на replay напад со session ID

Пример 1:

Имаме два клиенти во една безжична мрежа. Клиентот А сака да прати порака на клиентот Б. Клиентот А му праќа знак на клиентот Б. Тој знак клиентот Б го користи за да ја трансформира лозинката и како таков му го праќа на клиентот А. Од другата страна, клиентот А го прима знакот и го прави истото. Доколку двете вредности се исти, тогаш влезот е успешен. На тој начин преносот на податоци неможе да се повтори или пак да се загуби - заштитен е со знак. На овој начин се спречува replay нападот [2].

Пример 2:

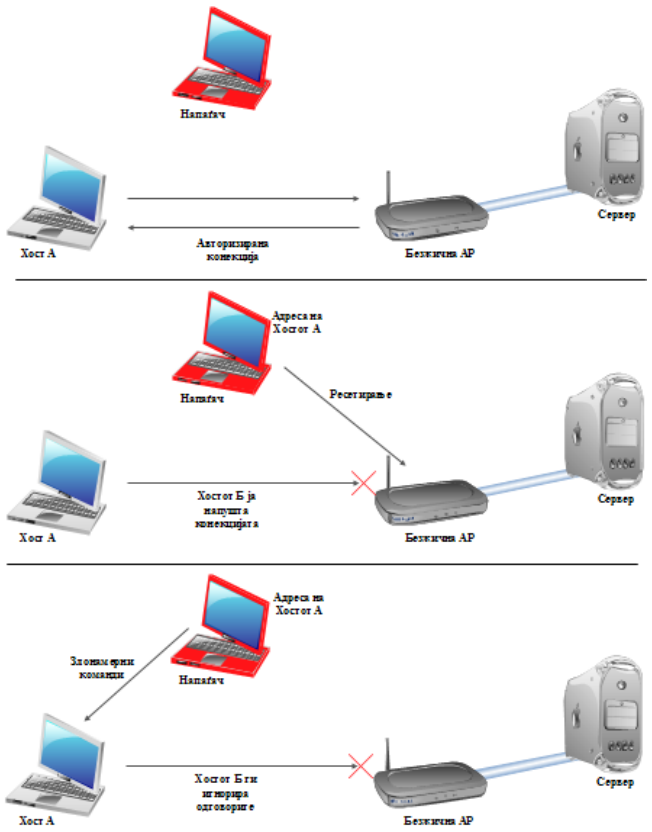
Да претпоставиме дека постои и трет клиент Ц. Тој клиент успеал некако да ја зачува оваа вредност и сака да ја употреби во некоја друга сесија. Во тој случај клиентот А праќа друг знак на сесија и кога клиентот Ц ќе ја трансформира лозинката со стариот знак, тогаш вредноста ќе биде различна од вредноста која што ќе ја има клиентот А.

Знаците на сесиите мора да бидат одбирани случајно и за секоја сесија да се користи различен знак. Инаку, од примерите погоре, клиентот Ц може да се претстави како клиент А, затоа што го знае неговиот знак и може лажно да се претстави кај клиентот Б и да го користи знакот на клиентот А. Клиентот Ц тогаш ќе може подоцна да врати на барањето од клиентот Б или пак воопшто да не врати и на тој начин да го изврши нападот. Клиентот А, исто така, има можност да користи некои посебни случаеви за праќање на пораки, но тоа вклучува користење на т.н, код за автентикација на пораката. Во тој случај проверката би ја вршел клиентот Б [3].

Важна работа е да се знае дека, доколку не се спречи нападот, тој може цело време да се повторува и да упаѓа во пораките. Во овој случај важна е улогата на идентификаторот кој ја идентификува пораката од клиентот до серверот со посебна ознака која само тие ќе ја знаат. Овие ознаки се користат дури и кога имаме силна шема на автентикација.

Овие знаци на сесии од најголемо значење се при транспорт на податоци во бежичната комуникација, кога комуникациската инфра-структура користи http протокол. Сите знаци на сесиите на секој корисник треба да бидат уникатни, непредвидливи и отпорни, сè со цел да не дозволат да се случи некој напад [4]. Нивната функција е да обезбедат што поголема безбедност при преносот на пораките и да бидат во состојба да го спречат replay нападот, како и т.н. киднапирање на порака (hijacking). Токму поради тоа се користи т.н. алгоритам за добивање на овие знаци. Карактеристично, а и една од најважните работи кај овој алгоритам, е тоа што при правењето на овие знаци на сесија не користи никакви лични податоци за корисникот, како на пример име, презиме, корисничко име, лозинка и слично. Овој алгоритам не јавува дупликат вредност на знакот на сесијата [5].

И покрај целата заштита која ја нудат овие идентификатори на сесиите, сепак постои ризик од некои пропусти, со што е возможно да се реализира одреден напад. Еден таков пример е нападот со киднапирање на сесија (Session Hijacking attack).



Слика 4 : Шематски приказа на hijacking

Тој се случува на тој начин што се врши истражување на целиот механизам на веб сесиите, сè со цел да може да ја земе ознаката. Затоа што http комуникацијата користи различни конекции, веб услугата има потреба од посебен метод за препознавање на секоја можна конекција на корисникот. Најкорисниот метод зависи од знакот кој веб услугата го праќа на клиентот после успешната автентикација на клиентот. Нападот со киднапирањето на сесијата го зема знакот со крадење или претпоставување на валидниот знак на сесијата, сè со цел да се здобие со влез до веб услугата. Токму поради ова, ознаката на сесијата, и покрај тоа што е една од најкористените методи за превенција од напади, има свои недостатоци и не нуди целосна безбедност.[5, 6].

B. Временска ознака (timestamps)

Временската ознака претставува низа од карактери кои го означуваат датумот или времето на системот кога се случил некој настан. Оваа ознака оди заедно со порака, со основна цел да го намали ефектот од replay нападите. Кај веб услугите, replay нападот се случува кога барањето преку http протоколот е прекинато, односно пресретнато и содржината е препратена до провајдерот во неговата оригинална форма [7].

Пример:

Клиентот А периодично го емитува своето време на часовникот заедно со неговата MAC адреса. Кога клиентот Б сака да му прати некоја порака на клиентот А, таа во својата порака ја запишува нејзината најдобра проценка за времето на неговиот часовник. Боб ја прима пораката само доколку временската ознака е во разумна толеранција.

Кога како метод за превенција од replay нападите се користи временската ознака, потребно е да се обрне внимание на неколку работи. Кога ставаме временска ознака во пораката, најпрво мора да направиме заштита на интегритетот на пораката со користење на безбеден транспорт, како на пример SSL (secure sockets layer) или пак да ја зголемиме безбедноста на ниво на порака, како на пример XML дигитален потпис и слично. Ако не вршиме заштита на интегритетот на временската ознака, постои можност да се зачува пораката и нејзината содржина да се препрати со различна, односно поинаква временска ознака, или пак како порака со поминат датум или пак да се искористат и двата претходни случаи заедно.

Најчеста вредност за искористената временска ознака е времетраење од 5 минути. Оваа вредност е карактеристична за истекот на времето на една порака која се испраќа до примачот, доколку не е назначена некоја друга вредност. Доколку постои некој различен, посебен истек за специфичен клиент или пак не сме сигурни за вредноста на таргет услугите, се прави конфигурација на вредноста на времето кое е потребно за пораката и се дава нова вредност за истото.

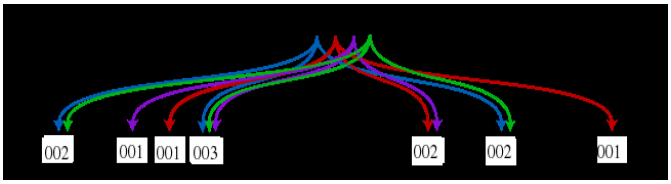
Временската ознака може да се користи и за да ја заштити пораката со идентификација на поминатото време кога пораката не е повеќе валидна. Валидацијата е постигната од страна на веб услугата кој ја прима пораката. За да се пресмета временската разликата помеѓу клиентот и серверот, може да се конфигурира часовникот и да се пресмета. Всушност, временската ознака може да се додаде на било која порака и тоа на многу едноставен начин. Тоа се прави со помош на елементот <add-timestamp>. На овој начин се зголемува безбедноста на пораката и ризикот од нејзин напад. Постои можност и да се изврши проверка на временската ознака и тоа се врши со помош на елементот <verify-timestamp> кој се наоѓа на самата порака. Овој елемент има два атрибута, односно јавува две пораки и тоа “истечен” и “создаден”. Ако се констатира дека креираниот атрибут е вистинит, тогаш се проверува временската ознака на пораката за да се види дали го содржи времето на создавање. Ако времето на создавање го нема. тогаш пораката е одбиена. Доколку го има, во тој случај пораката е примена.[8]

Временска ознака претставува еден од неколкуте методи кој се користи за превенција на replay нападите. Кога се користи овој метод синхронизацијата на времето треба да биде постигнато со користење на безбедносен протокол. Предноста на овој метод е тоа што серверот нема потреба да генерира различни броеви, односно знаци на сесиите.

C. Bloom Филтер

Методот кој најмногу се користи за намалување од replay нападите во бежичните мрежи е метод на т.н Bloom филтер. Bloom филтерот претставува високо ефикасен начин на вршење на заштита од напади на тој начин што

прави сет од изрази и ги складира во посебен филтер [9]. На секој израз може да се дадат два одговори за тоа дали изразот се наоѓа во филтерот. Одговорите се “веројатно е елемент” или пак ‘дефинитивно е елемент’. На тој начин се создава филтерот и лесно се врши проверка за тоа дали податоците се пренесуваат правилно или постои можност за напад. Bloom филтерот се состои од низа од неколку битови, означени како на пр. b1, b2…,bK, заедно со ≥ 1 hash функции f1, . . . , fn. Hush функциите се избираат случајно и си добиваат одредена вредност. Филтерот е празен кога сите вредности на битови се 0. Кога се дава барање до филтерот се стават битови во индексите.



Слика 5: Распоред на битовите кај bloom филтерот

Пример:

Добиваме едно барање. Ова барање најпрво го прима Bloom филтерот. Откога ќе го прими се поставуваме прашањето дали ова барање го има во себе или не. Доколку го има одговара со изразот “веројатно” , во спротивно јавува порака “дефинитивно го нема” [9].

Овој е принципот на кој функционира bloom филтерот. Значаен елемент кај овој метод е тоа што Bloom филтерот чува листа од веќе поминатите барања. Кога ќе пристигне ново барање, дискот прво ќе направи проверка во филтерот. Ако филтерот јави дека го нема тоа барање, дискот може безбедно да продолжи со пренос на пораката. Доколку јави дека тоа барање е веќе поминато, тогаш се праќа одбиена порака. Клиентот продолжува да праќа барања се додека не прими одобрување или одбиена порака за тоа барање. Ако се добие повратна одбивна порака, се менува дадениот случај на барањето и се продолжува со преносот.

Доколку филтерот се исполни со премногу барања, потребно е да имаме и друг таков резервен филтер. Резервниот филтер настанува со имплементација, одржување и спојување на неколку филтри заедно, при што секој е со посебен растечки број, и овие броеви периодично се проверуваат. Ова функционира на таков начин што секој филтер е поврзан со претходниот период (epoch) [10]. Кога на филтерот му треба замена, дискот го бара следниот број на периодот, го брише стариот филтер и го остава новиот филтер да ја врши работата. На рестартирање, бројот на периодот е зголемен од бројот на филтери. Ова овозможува праќање на порака додека системот е изгасен.

Слика 6: Приказ на работата на bloom филтерот

Клиентот праќа барање со некој број за којшто мисли дека е последниот. Секоја порака содржи одреден број, односно вредност. Доколку барањето што го праќа клиентот е со некој постар број, тогаш пораката се одбива. Во спротивно се проверува бројот на барањето во соодветниот bloom филтер. [11].

Методот за заштита од replay нападите со bloom филтерот е едноставен и скромен. Тој е еден од најискористените методи и истиот нуди најголема заштита од овие напади. Негативната страна му е тоа што работи на посложени системи.

IV. Методи за намалување на ефектот на replay нападите

При мрежно комуницирање помеѓу два клиенти или пак клиент со сервер, можно е да дојде до нецелосен пренос на пораките, дуплирање на податокот или пак намалување на истиот. Тој процес се нарекува replay напад. Replay нападот претставува напад на податоците што се пренесуваат каде што хакерот користи анализатор на протоколот и истиот ги мониторира и копира пакетите, како што поминуваат низ мрежата. Кога хакерот ќе го земе пакетот тој може да го филтрира, да ги менува податоците и покрај тоа што содржат дигитални потписи и други заштитни кодови. Откако пакетот со податоци е сменет, хакерот тој пакет го враќа во мрежата, испраќајќи го кон посакуваната дестинација.

Ваквиот напад е доста ризичен и токму поради тоа постојат одредени методи кои се користат за намалување на ефектот од него. Како најкористени методи се т.н знаци на сесии, временски ознаки и bloom филтерот. Знакот на сесијата претставува метод кога секоја сесија се карактеризира со посебен знак и секој пакет со податоци

што се праќа го носи тој знак. Овој метод нуди безбедност, меѓутоа не целосна и не комплетна. Временските ознаки се слични како знаците на сесиите, само што овие вршат временски запис на секој настан што се случил во системот. На тој начин имаат контрола над сите податоци кога поминуваат и кога треба да стигнат. Според истражувањата и детални опис наведен во овој труд, може да се каже дека овој метод би бил многу корисен за идентификување на нападот веднаш откако ќе се случи, но не е толку способен да го спречи овој напад. За разлика од овие два метода, најбезбеден начин да се спречи replay методот кај бежичните мрежи е со помош на bloom филтерот. Начинот на кој што ги проверува податоците е брз, едноставен, лесен и практичен. Така, тој лесно може да го примети нападот и да превземе соодветна мерка. Bloom филтерот е еден од најкорисните методи за превенција од replay нападите. Како најголема предност кај него е тоа што има резервен филтер, односно, доколку се наполни филтерот со податоци, постои друг филтер каде што ќе може да се складираат останатите податоци.

V. Заклучок

Replay нападот е еден од најчестите напади кој се случува при размена на податоци во бежичната мрежа. Со сè поголемата искористеност на безжичните мрежи можноста за реализирање на овој напад е уште поголема, пред сè поради полесната пристапност кон безжичниот медиум. Досега е направено многу во намалување на ефектите на овој напад, како и негово целосно превенирање.

Во овој труд се разгледани три механизми за заштита од replay нападот, и направена е споредба на ефикасноста на овие механизми. Како најсоодветен механизам против reply нападите од разгледаните се третира методот на заштита со употреба на Bloom филтер, кој, за разлика од останатите механизми кои се разгледани во овој труд, не само што реактивно се справува со replay нападите, туку има и проактивен карактер, т.е. истиот не само што може да го евидентира нападот откако ќе се случи, туку може целосно да го спречи..

Благодарност

Голема благодарност до професорот Митко Богдановски кој многу помогна во изработка на овој труд. Исто така благодарност сакаме да искажеме и до Европскиот Универзитет на РМ кој овозможува случување на вакви конференции и ни дозволува и ние да земеме учество и да бидеме дел од нив.

Библиографија

[1] Wheeler, E., Replay attacks, SANS Study guide, 2008.
[2] Gopal's blog, "Replay attack & Its countermeasures", July 30,2009
Available:http://joginipally.blogspot.com/2009/07/replay-attack-its-countermeasures.html
[3] OracleApplication Server, Web Services Security Guide, B28976-01, 2006.
Available:http://download.oracle.com/docs/cd/B32110_01/web.1013/b28976.pdf
[4] A Guide to Building Secure Web ApplicationsManaging, User Sessions-Chapter 7
Available:http://www.cgisecurity.com/owasp/html/ch07s02.html
[5] David Endler, IDEFENSE Labs, "Brute-Force Exploitations of Web Aplication Session IDs", November 1, 2001
Available:http://www.cgisecurity.com/lib/SessionIDs.pdf
[6] Shray Kapoor, "Session Hijacking, Ecploting TCP, UDP and HTTP Sessions"
Available:http://www.infosecwriters.com/text_resources/pdf/SKapoor_SessionHijacking.pdf
[7] Sreekanth Malladi, Jim Alves-Foss, Robert B.Heckondorn, "On Preventing Replay Attacks on Security Protocol", University of Idaho
Available: http://www2.cs.uidaho.edu/~jimaf/papers/replay02.pdf
[8] HASINI'S VIEWS, "Timestamp in WS-Security to mitigate replay attacks", SATURDAY, FEBRUARY 25, 2012
Available: http://hasinigunasinghe.blogspot.com/2012/02/timestamp-in-ws-security-to-mitigate.html
[9] Mark Lillibridge, "*Preventing Replay Attack*", 2003-01-06
Available:http://www.usenix.org/events/fast03/tech/full_papers/aguilera/aguilera_html/node7.html
[10] Christian Antognini, Trivadis AG, Zurisch, Switzeraland, "Bloom Filters", juni 2008
Available:http://antognini.ch/papers/BloomFilters20080620.pdf
[11] Ip research & Communities, "System and method for preventing replay attacks"
Available:http://freepatentsonline.com/y2005/0022009.html