HRVATSKA AKADEMSKA I ISTRAŽIVAČKA MREŽA
CROATIAN ACADEMIC AND RESEARCH NETWORK

Analiza RADIUS protokola

CCERT-PUBDOC-2001-07-05

CARNet CERT u suradnji s LS&S

Sigurnosni problemi u računalnim programima i operativnim sustavima područje je na kojem CARNet CERT kontinuirano radi.

Rezultat toga rada ovaj je dokument koji je nastao suradnjom CARNet CERT-a i LS&S-a, a za koji se nadamo se da će Vam koristiti u poboljšanju sigurnosti Vašeg sustava.

CARNet CERT, www.cert.hr - nacionalno središte za **sigurnost računalnih mreža** i sustava.

LS&S, www.lss.hr - laboratorij za sustave i signale pri Zavodu za elektroničke sustave i obradbu informacija Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu.

Sadržaj

1. Uvod

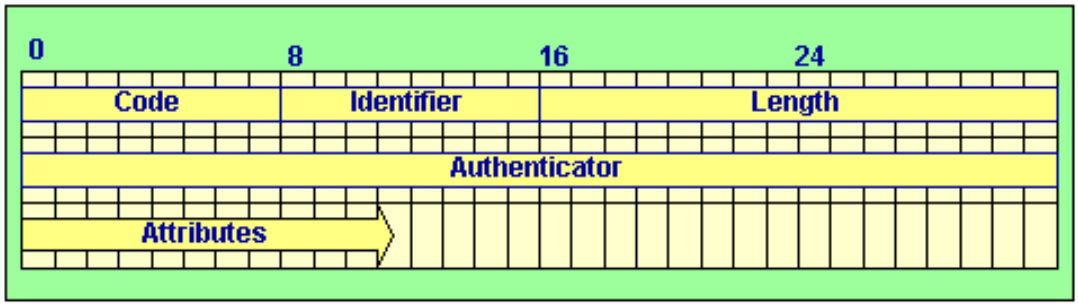
RADIUS je danas vrlo često korišten protokol za autentikaciju, autorizaciju te administraciju korisnika, čija je primjena najraširenija upravo kod ugrađenih mrežnih uređaja poput usmjerivača, preklopnika, modema, te brojnih drugih njima sličnih uređaja.
Protokol se bazira na klijent-poslužitelj modelu *(engl. client-server)*, koji kao transportno sredstvo koristi UDP mrežni protokol.
Na strani klijenta koristi se **N**etwork **A**ccess **S**erver (NAS) programski paket, koji obavlja zadaće vezane za prosljeđivanja korisničkih parametara RADIUS poslužitelju, te obradu primljenih odgovora.
S druge strane, RADIUS poslužitelji zaduženi su za prihvaćanje upita, provjeru primljenih korisničkih parametara, te vraćanja potrebnih konfiguracijskih parametara, koji će klijentu omogućiti pružanje adekvatne usluge korisniku.
RADIUS poslužitelji se također mogu koristiti i kao proxy klijenti drugim RADIUS poslužiteljima, ili nekim drugim sustavima za autentikaciju korisnika.
Sama komunikacija između klijenta i poslužitelja bazira se na tajnom ključu kojeg dijele klijent i poslužitelj, i koji se kao takav iz sigurnosnih razloga nikada ne šalje računalnom mrežom.
RADIUS protokol koristi se iz više razloga :

- mrežni uređaji poput gore navedenih u osnovi ne posjeduju mogućnost pohranjivanja velikog broja autentikacijskih parametara različitih korisnika, s obzirom na ograničene resurse s kojima raspolažu.
- RADIUS protokol olakšava i centralizira administraciju korisnika, što može biti vrlo važan, te presudan čindbenik u okolinama gdje postoji potreba za administracijom velikog broja korisnika. Kao primjer mogu se uzeti veći ISP (Internet Service Provideri) davatelji usluga, kod kojih se svakodnevno javlja potreba za dodavanjem novih, brisanjem starih, te modificiranjem postojećih korisničkih računa. U takovim okolinama alati za centraliziranu i jednostavniju administraciju poput RADIUS protokola mogu imati presudni značaj na njihovo poslovanje.
- RADIUS protokol pruža određeni nivo zaštite protiv aktivnih napada neovlaštenih korisnika. Za razliku od većine drugih protokola ovog tipa koji nude ili nedovoljnu, ili povremenu ili uopće ne pružaju nikakvu zaštitu, RADIUS u tu svrhu koristi TACACS+ i LDAP protokole za autentikaciju.
- Velika podrška različitih proizvođača mrežne opreme. Budući da se RADIUS protokol implementira najčešće u sklopu ugrađenih mrežnih uređaja, u takovim okolinama postoji mala ili gotovo nikakva mogućnost nadogradnje drugih protokola. Zbog raširenosti ovog protokola svaka promjena koja bi se načinila u smjeru njegovog poboljšanja morala bi biti kompatibilna s postojećim rješenjima.

Upravo se iz navedenih razloga RADIUS protokol u današnje vrijeme smatra *de facto* standardom za udaljenu autentikaciju korisnika, te se kao takav implementira i kod novijih i kod starijih mrežnih uređaja.

2. RADIUS protokol

Podatci se između klijenta i poslužitelja razmjenjuju putem RADIUS podatkovnih paketa, enkapsuliranih unutar UDP paketa protokola niže razine. RADIUS komunikacija koristi upit-odgovor *(engl. challenge-response*) paradigmu, u kojoj klijent šalje upite poslužitelju, a poslužitelj na temelju njih vraća odgovor klijentu.
Format jednog RADIUS paketa je sljedeći:



Slika 1: *Format RADIUS podatkovnog paketa*

Značenje pojedinih polja:

- **Code –** jedan bajt veliko polje, koje definira tip RADIUS podatkovnog paketa. Moguće vrijednosti ovog polja prikazane su u sljedećoj tablici:

Vrijednost	Opis
1	Access-Request
2	Access-Accept
3	Access-Reject
4	Accounting-Request
5	Accounting-Response
11	Access-Challenge
12	Status-Server (experimental)
13	Status-Client (experimental)
255	Reserved

Slika 2: *Moguće vrijednosti Code polja Radius paketa*

- **Identifier (ID)-** jedan bajt veliko polje, koje klijentu omogućuje jednoznačnu identifikaciju parova upit-odgovor.
- **Length –** 2 bajta koja predstavljaju veličinu paketa.
- **Authenticator –** vrijednost koja se koristi za provjeru ispravnosti odgovora od strane RADIUS poslužitelja, a ujedno se koristi kao dio algoritma za prikrivanje, odnosno zaštitu korisničke zaporke.
- **Attributes** – sekcija u kojoj se nalaze proizvoljni atributi koji pripadaju samoj sesiji (upitu ili odgovoru). Jedini atributu koji su obavezni su `User-Name` i `User-Password` atributi, a ostali su proizvoljni.

U nastavku će biti iznesen jedan tipični postupak RADIUS autentikacije, u kojemu RADIUS klijent na temelju korisničkog zahtjeva poslužitelju šalje `Access-Request` upit sa navedenim korisničkim imenom i zaporkom, na što poslužitelj odgovara sa `Access-Accept`, `Access-Reject` odgovorom ili odbijanjem sesije.

U ovakvom pristupu klijent je taj koji želi svoje autentikacijske parametre (korisničko ime i zaporku) provjeriti kod poslužitelja, dok je poslužitelj onaj koji ima pravo pristupa sustavu baze podataka, sa arhivom korisničkih parametara autentikacije.

Na temelju postavljenog upita RADIUS klijenta, poslužitelj onda može obraditi korisnički zahtjev, te ovisno o regularnosti primljenih parametara dozvoliti ili zabraniti pristup resursima.

2.1. Upit klijenta

Klijent započinje sesiju kreiranjem RADIUS upita sa postavljenim `Access-Request` kodom, koji mora minimalno sadržavati `User-Name` i `User-Password` korisničke atribute.
Bajt identifikacije (ID) tog paketa odabire se proizvoljno od strane klijenta i kao takav nije definiran RADIUS protokolom. Generiranje ovog broja najčešće je implementirano u obliku jednostavnog brojila koje se uvećava za jedan generiranjem svakog novog upita.
Paket također sadrži `Request Authenticator` vrijednost unutar `Autenthicator` polja RADIUS paketa. Ova vrijednost također predstavlja slučajno odabrani 16 bajtni znakovni niz, a algoritam njegovog generiranja kako ćemo kasnije vidjeti, ima presudan značaj za sigurnost ovog protokola.
RADIUS paket sam po sebi nije zaštićen ni na koji način, izuzev `User-Password` atributa koji se iz sigurnosnih razloga zaštićuje enkripcijom na način opisan u nastavku dokumenta.
U ovakvom pristupu klijent i poslužitelj međusobno dijele neku vrstu tajnog ključa *(engl. secret),* koji predstavlja podlogu za postupak kriptiranja korisničke zaporke**.** Postupak zaštite zaporke je sljedeći:

- `Request Authenticator` vrijednost (RA) se spaja sa spomenutim tajnim ključem (S) kojeg dijele klijent i poslužitelj.
- tako dobiveni niz obrađuje se MD5 hash funkcijom, što će rezultirati 16 bajtnim znakovnim nizom
- slijedi primjena XOR funkcije između dobivenog 16 bajtnog niza i korisničke zaporke, kako bi se na taj način došlo do zaštićene zaporke. Ukoliko je korisnička zaporka duža od 16 bajtova, provode se dodatne MD5 kalkulacije, kako bi se došlo do željenog rezultata.

Formalnije*:*
Označimo tajni ključ klijenta i poslužitelja sa S, te pseudo slučajnu 128 bitnu vrijednost `Request Authenticator` s RA. Korisnička zaporka dijeli se na 16 bajtne blokove `p1,…,pn`, gdje će vrijednost zadnjeg bloka biti nadopunjena nulama, kako bi se dobilo točno 16 blokova veličine jednog bajta.
Opisanim matematičkim operacijama (izraz 1) dobivaju se `c1,…, cn` blokovi, čijim se kombiniranjem dobiva konačni rezultat.

```
c1 = p1 XOR MD5 (S + RA)
c2 = p2 XOR MD5 (S + c1)
.
.                                                    (1)
.
cn = pn XOR MD5 (S+cn-1)
```

Zaštićeni `User-Password` atribut dobit će se spajanjem dobivenih c1,…,cn blokova (izraz 2).

```
User-Password = c1 + c2 +… +cn                       (2)
```

, gdje operator + predstavlja operaciju spajanja znakovnih nizova.

2.2. Odgovor poslužitelja

Nakon primljenog RADIUS `Access-Request` upita, poslužitelj prvo provjerava da li za tog klijenta postoji tajni ključ koji oni međusobno dijele. Ukoliko poslužitelj ne posjeduje tajni ključ tog klijenta, zahtjev se odbija.
Ukoliko takav ključ postoji, poslužitelj prolazi kroz nešto izmijenjeni postupak kriptiranja, kako bi došao do originalne nezaštićene korisničke zaporke.
Nakon toga slijedi postupak provjere ispravnosti dobivene korisničke zaporke, koji će ukoliko je regularan rezultirati vračanjem `Access-Accept` paketa klijentu.
Ukoliko je proslijeđena zaporka neispravna (ne odgovara paru `Username/Password` koji poslužitelj posjeduje u svojoj bazi podataka) klijentu se vraća `Access-Reject` paket, kojim se odbija nastavak procesa autentikacije.

U oba slučaja `Access-Accept` i `Access-Reject` vraćeni paketi sadrže identičnu vrijednost bajta identifikacije (ID), kao što je sadržavao i originalni `Access-Request` upit klijenta.
Vrijednost `Response Authenticator` atributa vraćenog paketa dobiva se primjenom MD5 hash funkcije na parametre istog paketa u kombinaciji sa vrijednošću polja `Response Authenticator` originalnog upita klijenta (izraz 3).
Matematički formulirano:

```
RA = MD5(Code+ID+Length+RequestAuth+Attributes+S)        (3)
```

,gdje + operator označava operaciju spajanja.

2.3. Procesiranje odgovora poslužitelja

Nakon primljenog odgovora od strane RADIUS poslužitelja, klijent na temelju bajta identifikacije pokušava odrediti da li se odgovor zaista odnosi na njegov upit.
Uspoređivanjem vrijednosti ID polja poslanog (`Access-Accept`) i primljenog (`Access-Accept ili Access-Reject`) paketa klijent vrlo jednostavno može identificirati regularnost odgovora, odnosnu njegovu pripadnost upitu klijenta.
Ukoliko se ove dvije vrijednosti ne slažu, klijent odgovor smatra neregularnim i sesija se prekida.
Slijedi provjera `Response Authenticator` polja primljenog paketa provođenjem iste matematičke obrade kao i na strani poslužitelja, kako bi se potvrdilo da odgovor zaista stiže od strane poslužitelja.
Ukoliko se rezultati ne poklapaju, odgovor se smatra neregularnim i sesija se prekida.
Ukoliko je klijentu vraćen `Access-Accept` RADIUS paket sa valjanim sadržajem, korisničko ime i zaporka smatraju se regularnima, što će rezultirati uspješnom autentikacijom korisnika.
Ukoliko je klijentu vraćen `Access-Reject` RADIUS paket sa valjanim sadržajem, korisničko ime i zaporka smatraju se neregularnim, što će rezultirati neuspješnom autentikacijom korisnika.

3. Problemi sa RADIUS protokolom

RADIUS protokol sadrži niz sigurnosnih propusta, koji su posljedica ili propusta u implementaciji samog protokola ili neispravne, odnosno nepotpune implementacije programske podrške. U nastavku ovog dokumenta biti će opisani neki od sigurnosnih propusta vezanih za RADIUS protokol.

3.1. Napad na tajni ključ pomoću `Response Authenticator` polja

Podloga za provođenje ovog tipa napada na RADIUS protokol, je upravo priroda samog načina na koji se generira vrijednost `Response Authenticator` polja, kod `Access-Accept` ili `Access-Reject` RADIUS paketa.
Naime, pregledavanjem upravo navedenih RADIUS paketa, te korištenjem snažnih algoritama za razbijanje, postoje realne mogućnosti za probijanje vrijednosti tajnog ključa.
Ukoliko se pomnije promotre argumenti MD5 hash algoritma, koje poslužitelj koristi za generiranje odgovora klijentu (izraz 3), može se primijetiti da je jedina nepoznanica u tom izrazu upravo tajni ključ koji dijele klijent i poslužitelj.
Na temelju detektiranih RADIUS paketa, napadaču se postavlja ovakav problem:

```
RA = MD5(Code+ID+Length+RequestAuth+Attributes+X)        (4)
```

, gdje X predstavlja tajni ključ koji se želi odgonetnuti.

Na temelju poznatih argumenata napadač u svakom trenutku može izračunati slijedeći izraz:

```
RA = MD5(Code+ID+Length+RequestAuth+Attributes)     (5)
```

Sada se metodom pokušaja i pogreške, te snažnim računalnim algoritmima može s vrlo velikom vjerojatnošću odgonetnuti tajni ključ, klijenta i poslužitelja.

3.2. Enkripcija User-Password atributa

Algoritam koji se u slučaju RADIUS protokola koristi za ekripciju korisničke zaporke, odnosno `User-Password` atributa, spada u grupu *stream chiper* algoritama za enkripciju podataka, gdje se MD5 hash funkcija koristi kao generator pseudo slučajnih brojeva (PRNG).
Upravo sigurnost ovakvog postupka zaštite povjerljivih podataka (u ovom slučaju korisničke zaporke) ovisi o snazi i kvaliteti, u ovom slučaju odabrane MD5 hash funkcije, odnosno o kvaliteti odabira tajnog ključa kojeg dijele klijent i poslužitelj. Što su ovi elementi kvalitetnije odabrani, sigurnost cijelog postupka biti će na višoj razini.
No, budući da MD5 hash funkcija u svojoj osnovi nije predviđena da se koristi kao primitiva *stream chiper* grupe algoritma, već kao čisti alat za enkripciju podataka, ona se u slučaju RADIUS protokola smatra neprikladno odabranim alatom.
Kako će se vidjeti u nastavku dokumenta, upravo će spomenuto neadekvatno korištenje MD5 hash primitive u slučaju RADIUS protokola, biti uzrok određenog broja sigurnosnih propusta, koje neovlašteni korisnik može iskoristiti za neautorizirani pristup povjerljivim korisničkim podatcima.

3.3. Napad na tajni ključ pomoću `User-Password` atributa

Kao posljedica korištenja *stream chiper* algoritma za kriptiranje korisničke zaporke prilikom slanja upita poslužitelju, napadaču se i u ovom slučaju pruža mogućnost odgonetanja tajnog ključa kojeg dijele klijent i poslužitelj.
Napad počinje pokušajem autentikacije napadača kod RADIUS klijenta, sa njemu poznatom, i u tu svrhu osmišljenom korisničkom zaporkom. Na temelju tako primljenog zahtjeva RADIUS klijent formirat će `Access-Request` upit, koji će se proslijediti poslužitelju u svrhu provjere korisničkih podataka.
`User-Passsword` atribut tog paketa, biti će dobiven na način kako je to opisano u poglavlju 2.1, sa naglaskom na činjenicu da je kao argument enkripcije korištena napadaču poznata zaporka.
Ukoliko sada isti napadač analizom mrežnog prometa uhvati generirani RADIUS `Access-Request` upit, te primjeni XOR operaciju na zaštićeni `User-Password` atribut, kao rezultat dobit će izlaznu vrijednost slijedeće matematičke operacije:

```
MD5(Shared Secret + Request Authenticator)                (6)
```

S obzirom da je vrijednost `Request Authenticator` polja poznata, i da se ista može naći unutar `Access-Request` paketa, ostaje samo jedna nepoznanica, i to upravo tajni ključ koji se želi odgonetnuti.
Metodom pokušaja i pogreške te uporabom snažnih računalnih algoritama, i na ovaj način je moguće doći do povjerljivog tajnog ključa klijenta.

3.4. Napad korisničku zaporku pomoću `User-Password` atributa

Drugi sigurnosni propust koji je također posljedica neprikladne upotrebe *stream chiper* algoritma za kriptiranje korisničke zaporke, neovlaštenom korisniku omogućuje uspješno nagađanje korisničke zaporke, a samim time i uspješnu autentikaciju kod poslužitelja.
Osnovni preduvjet za provođenje ove vrste napada je taj da poslužitelj ne posjeduje ograničenje na maksimalni broj neuspjelih pokušaja autentikacije.
Prvi dio napada isti je kao i u prethodnom slučaju. Napadač klijentu upućuje zahtjev za autentikacijom sa valjanim korisničkim imenom proizvoljnog korisnika, te sa nasumce predviđenom njegovom zaporkom (najvjerojatnije pogrešnom).
Nakon tog potrebno je na način opisan u prethodnom poglavlju (3.3), doći do izlazne vrijednosti matematičke operacije

```
MD5(Shared Secret + Request Authenticator)                (7)
```

Ukoliko se promotri izraz (1) može se primijetiti da napadač na temelju ovako izračunate vrijednosti, može u svakom trenutku generirati novi `Access-Request` paket sa istim korisničkim imenom i drugom, novom, opet nasumce pretpostavljenom vrijednošću korisničke zaporke.
Ukoliko poslužitelj nema definirano ograničenje na broj pokušaja prijavljivanja u sustav, napadač slanjem velikog broja `Access-Request` paketa sa nasumce odabranom vrijednošću korisničke zaporke, može doći do ispravne korisničke zaporke.

3.5. Napadi bazirani na `Response Authenticator` atributu

Kompletna sigurnost RADIUS protokola u osnovi ovisi o kvaliteti algoritma za generiranje `Request Authenticator` atributa. U svrhu uspješnog i sigurnog funkcioniranja RADIUS protokola spomenuti atribut mora biti jedinstven i nepredvidljiv.
Budući da sama specifikacija ovog protokola izričito ne ukazuje na izuzetnu važnost postupka generiranja ovog atributa, određeni broj implementacija koristi loše i površne implementacije mehanizma za generiranje slučajnih brojeva, koji se koristi u svrhu generiranja vrijednosti ovog atributa.
Naime, poznato je da kvaliteta i sigurnost gotovo kod svih algoritama za kriptiranje ponajviše ovisi upravo o mehanizmu za generiranje slučajnih brojeva *(engl. pseudorandom number generator, ili PRNG).*
Što više taj mehanizam posjeduje deterministička svojstva, to je algoritam enkripcije lakše provaliti. Potpuno identična situacija je i sa RADIUS protokolom.

3.5.1. Pasivni napad pomoću `Request Authenticators` atributa

Pasivnim promatranjem mrežnog prometa između RADIUS klijenta i poslužitelja, napadač s vremenom može kreirati neku vrstu RADIUS rječnika sa `Request Authenticators` atributima i njima odgovarajućim zaštićenim `User-Password` atributima.
Promatranjem veće količine mrežnog prometa, napadaču se otvara mogućnost da iz kriptiranih korisničkih zaporki eliminira utjecaj tajnog ključa, te da na taj način dođe do originalne nezaštićene korisničke zaporke.
Naime, primjenom XOR logičkog operatora nad zaštićenim korisničkim zaporkama, napadač kao rezultat može dobiti XOR kombinaciju nezaštićenih zaporki, što predstavlja prvi korak napada.
Uspješnost ovog napada uvelike će ovisiti o svojstvima korisničkih zaporki. Ukoliko su one sve jednake duljine ovaj napad neće dati nešto značajnije rezultate.
No, budući da praksa pokazuje da su korisničke zaporke u pravilu različitih duljina i svojstava, ova vrsta napada će u većini slučajeva rezultirati većim ili manjim uspjehom.
Idealna situacija za uspješno provođenje opisanog napada je kada su zaporke kraće od 16 bajtova, te međusobno različitih duljina.
Raznim statističkim metodama, te uzastopnim ponavljanjem pokušaja napada neovlaštenom korisniku se ovim putem otvara mogućnost otkrivanja korisničke zaporke korisnika.

3.5.2. Aktivni napad pomoću `Request Authenticators` atributa

U ovom slučaju neovlašteni korisnik započinje napad slanjem većeg broja RADIUS zahtjeva klijentu, sa svojim proizvoljno odabranim korisničkim zaporkama, čime će se aktivirati slanje `Access-Request` upita poslužitelju.
Nakon toga slijedi presretanje generiranih paketa prema poslužitelju, te bilježenje `Request Authenticator` i `User-Password` atributa unutar tih paketa .
I u ovom slučaju potrebno je uočiti da se u ovim paketima nalazi zaštićena zaporka, koja je poznata napadaču, budući da je on inicirao sam upit.
Primjenom XOR logičkog operatora između poznatih nezaštićenih, te generiranih zaštićenih zaporki, napadač vrlo jednostavno dolazi do `MD5(Shared Secret + Request Authenticator)` vrijednosti generiranih `Access-Request` paketa.
Na ovaj način napadaču se omogućuje kreiranje rječnika sa parovima `Request Authenticator` atributa i njima pripadnih `MD5(Shared Secret + Request Authenticator)` vrijednosti.

Ukoliko neovlašteni korisnik sada detektira regularni `Access-Request` upit sa nekom od vrijednosti `Request Authenticator` atributa koja je zabilježena u prethodnoj fazi napada, i u ovom slučaju postoji realna mogućnost dolaska do korisničke zaporke.
Naime, primjenom XOR operacije između `MD5(Shared Secret + Request Authenticator)` vrijednosti koja prema kreiranom rječniku odgovara uočenom `Request Authenticator` atributu, i zaštićenog `User-Password` polja detektiranog paketa, moguć je dolazak do valjane nezaštićene korisničke zaporke.

3.5.3. Krivotvorenje odgovora temelju `Request Authenticator` atributa

Neovlašteni korisnik također može praćenjem i analizom mrežnog prometa kreirati rječnik sa odgovarajućim `Request Authenticators`, ID vrijednostima, te odgovarajućim odgovorima poslužitelja.
Ukoliko sada neovlašteni korisnik između klijenta i poslužitelja detektira RADIUS paket sa vrijednostima ID i `Request Authenticators` polja, koje se poklapaju sa nekom od vrijednosti zabilježenih u prethodnoj fazi napada (kreiranje odgovarajućeg rječnika vrijednosti), napadač se može lažno predstaviti klijentu kao poslužitelj, te vratiti odgovor koji prema stvorenom rječniku pripada detektiranom paru vrijednosti.
Na ovaj način napadaču se omogućuje regularna autentikacija kod RADIUS poslužitelja, bez poznavanja valjane korisničke zaporke.

3.5.4. Provođenje DoS-a predviđanjem vrijednosti `Request Authenticator` atributa

Podloga za provođenje napada uskraćivanjem računalnih resursa u ovom slučaju upravo je eventualna predikcija generiranih vrijednosti `Request Authenticator` atributa.
Naime, napadač privremeno može preuzeti ulogu RADIUS klijenta, te na temelju predviđenih vrijednosti `Request Authenticator i` ID polja započeti slanje regularnih `Access-Request` upita.
Zbog nepoznavanja ostalih parametara `Access-Request` paketa, (korisničkog imena i zaporke) poslužitelj će na ove upite odgovarati `Access-Reject` paketima.
I u ovoj situaciji će napadač kreirati prikladni rječnik vrijednosti koji će povezivati maliciozno generirane `Access-Request` upite, te njima odgovarajuće `Access-Reject` odgovore.
Na temelju ovako kreiranog rječnika napadač će u fazi provođenja stvarnog napada moći na regularne i valjane `Access-Request` upite klijenta odgovarati lažnim (ali regularnim) `Access-Request` odgovorima iz rječnika, što će onemogućiti autentiakciju korisnika sa valjanim korisničkim imenom i zaporkom.

3.6. Napomene vezane za tajni ključ

Specifikacija RADIOUS protokola dozvoljava korištenje istog tajnog ključa za više korisnika RADIOUS sustava. Iako to nije eksplicitno izneseno, takva praksa smatra se lošim rješenjem i ne preporučuje se u ni u kojim slučajevima.
Naime, takav pristup dodatno međusobno povezuje sve korisnike, što napadaču olakšava provođenje svih gore opisanih napada na RADIUS protokol, budući da smanjuje broj pokušaja potrebnih za uspješno provođenje napada.
Dodatni problem je taj što većina implementacija RADIOUS klijenta i poslužitelja za tajni ključ dozvoljavaju korištenje samo ASCII znakovnih nizova, što unosi dodatni sigurnosni rizik.
Naime na taj način moguće je korištenje samo 94 znaka iz skupa ASCII znakova(od njih 256), što napadaču također bitno olakšava zadatak.
Osim toga neke implementacije unose dodatno ograničenje na tajni ključ, time što njegovu duljinu ograničavaju na svega 16 znakova ili manje.

4. Zaključak

4.1. Sažetak razmatranja

Iz upravo iznesenog izlaganja može se zaključiti da korištenje RADIUS protokola povlači nekoliko sigurnosnih pitanja, koja su posljedica samog dizajna i implementacije protokola. Nekoliko je elemenata uzrok tomu:

- neadekvatan odabir algoritma za zaštitu korisničke zaporke. Naime, u svrhu enkripcije korisničke zaporke nikako se ne bi se smjelo koristiti postupak iz *stream chiper* grupe algoritama, kao što je to u ovom slučaju. Posljedica ovog propusta su problemi opisani u poglavljima 3.2, *3.3, 3.4, 3.5.1, 3.5.2.*
- loša implementacija algoritma generiranja `Request Authenticator` atributa, iako je kao ideja ovo ispravan pristup (problem opisan u poglavlju 3.1).
- Slanje `Access-Request` upita poslužitelju ne sadrži nikakav element autentikacije klijenta kod poslužitelja (problem opisan u poglavlju 3.4).
- Prediktivnost generatora slučajnih brojeva za generiranje `Request Authenticators` atributa (problem opisan u poglavlju 3.5)
- Nedovoljno kvalitetan odabir tajnog ključa od strane administratora sustava. Dodatna ograničenja na odabir ključa, olakšavaju postupak njegovog razbijanja. (problem opisan u poglavlju 3.6)

4.2. Prijedlozi za poboljšanje protokola

Odabir novog dobro poznatog simetričnog *block chiper* algoritma za enkripciju korisničke zaporke, bilo bi dobro rješenje. Uvođenje novog `User-Password` atributa sa alternativnim algoritmom za enkripciju (npr. TDES).

Pažljiviji odabir ključa za enkripciju korisničke zaporke, nezavisno od tajnog ključa. Opcija je da se kao ključ za enkripciju koristi kombinacija tajnog ključa i `Request Authenticators` atributa.

4.3. Prijedlozi za modifikaciju RADIUS klijenta

Sama specifikacija RADIUS protokola trebala bi zahtijevati korištenje snažnijih i moćnijih generatora slučajnih brojeva (PRNG), u svrhu generiranja `Request Authenticators` atributa (npr. ANSI X9.17 specifikacija generatora slučajnih brojeva) .

Za svakog klijenta preporučuje se korištenje drugačijeg tajnog ključa, u obliku slučajnog znakovnog niza minimalne duljine 16 znakova, kojeg bi također generirao kvalitetan generator slučajnih brojeva.

Ukoliko ne postoji realna mogućnost unošenja značajnijih promjena unutar same specifikacije RADIUS protokola zbog problema kompatibilnosti, uvijek postoji mogućnost unošenja manjih preinaka i poboljšanja, koja bi unaprijedila ovaj protokol, te ga učinila sigurnijim, a ujedno bi se zadržala kompatibilnost sa starijim inačicama.

4.4. Zašto unositi promjene?

Osnovno pitanje koje se postavlja u većini slučajeva je, zašto uopće unositi promjene u RADIUS protokol? Zašto ne ići na neki noviji, napredniji, sigurniji i kvalitetniji protokol, koji bi u određenoj mjeri riješio probleme opisane u ovom dokumentu. Odgovor na to pitanje je, zato, jer takav protokol trenutno ne postoji!

Trenutno ne postoji takav protokol koji bi zamijenio RADIUS, a koji bi ujedno bio savršen, i koji bi riješio sve ranije spomenute probleme.

Eventualni kandidat koji bi u skorije vrijeme mogao predstavljati kvalitetnije rješenje u ovom području je Diameter protocol (IETF). U tom pogledu najveći dio poboljšanja usmjeren je upravo ka uklanjanju propusta u specifikaciji samog protokola, te funkcionalnih ograničenja koje on nameće.

No, također treba napomenuti da je malo toga napravljeno u smjeru poboljšanja sigurnosti same komunikacije između klijenta i poslužitelja. Naprotiv, u slučaju Diameter protokola iz same specifikacije uklonjene su bilo kakve funkcionalnosti vezane za samu sigurnost, što se u ovom slučaju ostavlja drugim mrežnim sigurnosnim protokolima.

U svrhu pojašnjenja ove tvrdnje priložen je izvadak iz same specifikacije Diameter protokola, koji pobliže opisuje ovu tvrdnju:
"Diameter clients, such as Network Access Servers (NASes) and Foreign Agents MUST support IP Security, and MAY support TLS. Diameter servers MUST support TLS, but the administrator MAY opt to configure IPSec instead of using TLS. Operating the Diameter protocol without any security mechanism is not recommended."
Iz ovoga dijela može se zaključiti da se nikako ne preporučuje samostalno korištenje Diameter protokola, upravo iz razloga što on sam po sebi neće pružati nikakav mehanizam zaštite. Taj dio prepušten je drugim mrežnim sigurnosnim protokolima koje će morati podržavati Diameter klijenti i poslužitelji, kako bi se postigao zadovoljavajući nivo sigurnosti.
Protokoli koje specifikacija u tu svrhu definira su IPsec za Diameter klijenta, odnosno IPSec i/ili TLS za Diameter poslužitelj. Ovakav pristup može se smatrati odličnom idejom budući da su spomenuti protokoli dizajnirani upravo u tu svrhu, te da su vrlo dobro poznati i pregledani od velikog broja korisnika u svrhu njihovog poboljšanja.
Nedostatak ovog pristupa je taj što će dodatno korištenje sigurnosnih protokola (IPSec, TLS) u kombinaciji sa Diameter protokolom, predstavlajti dodatne zahtjeve za mrežne računalne resurse. Naime, potpuna implementacija spomenutih sigurnosnih protokola, mogla bi naići na ozbiljna ograničenja kod većine današnjih ugrađenih mrežnih uređaja, što bi moglo predstavljati uzrok novih sigurnosnih problema.
Upravo ova ograničenja će još neko vrijeme biti uzrok korištenja RADIUS protokola. No postoji mogućnost u kojoj će proizvođači prijeći na površnu i nepotpunu implementaciju ovog protokola, što bi moglo predstavljati izvor novih problema.