HRVATSKA AKADEMSKA I ISTRAŽIVAČKA MREŽA
CROATIAN ACADEMIC AND RESEARCH NETWORK

SMTP protokol

CCERT-PUBDOC-2006-05-159

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 4

2. OSNOVE SMTP PROTOKOLA 5

2.1. SMTP MODEL 5
2.2. SMTP PROCEDURE I NAREDBE 6
2.2.1. EHLO / HELO naredba 6
2.2.2. MAIL naredba 6
2.2.3. RCPT naredba 7
2.2.4. DATA naredba 7
2.2.5. Ostale osnovne SMTP naredbe 8

3. EKSTENZIJE SMTP PROTOKOLA 8

4. SMTP AUTENTIKACIJA 10

4.1. AUTH NAREDBA 10

5. METODE SMTP AUTENTIKACIJE 11

5.1. LOGIN AUTENTIKACIJA 11
5.1.1. BASE64 kodiranje 11
5.2. PLAIN AUTENTIKACIJA 12
5.3. ANONIMNA AUTENTIKACIJA 13
5.4. CRAM-MD5 AUTENTIKACIJA 13
5.5. KERBEROS V4 AUTENTIKACIJA 14
5.6. GSSAPI AUTENTIKACIJA 15
5.7. S/KEY AUTENTIKACIJA 16
5.8. NTLM AUTENTIKACIJA 16

6. SIGURNOST SMTP PROTOKOLA 17

7. ZAKLJUČAK 19

8. REFERENCE 19

1. Uvod

SMTP (eng. *Simple Message Transfer Protocol*) je komunikacijski protokol čija je osnovna namjena siguran i pouzdan prijenos poruka elektroničke pošte (eng. *e-mail*) bez obzira na tehnologiju i sustav koji se koristi za ostvarenje samog prijenosa. SMTP protokol je relativno jednostavan tekstualni protokol koji se počinje koristiti 1980. godine kada postupno zamjenjuje UUCP (eng. *Unix to Unix CoPy Protocol*) protokol koji je prilagođen prijenosu poruka elektroničke pošte u situacijama kada su primatelj i pošiljatelj samo povremeno spojeni na komunikacijsku mrežu, dok se SMTP protokol pokazao bolji u situacijama kad su primatelj i pošiljatelj konstantno povezani na komunikacijsku mrežu.

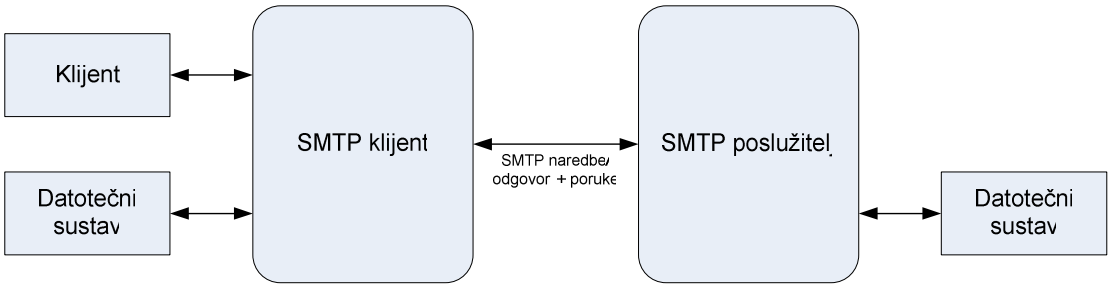
Pošto je SMTP protokol baziran na ASCII tekstualnim naredbama, ubrzo nakon početka njegove primjene na vidjelo su izašla ograničenja koja su time nametnuta pa se pristupilo definiranju ekstenzija SMTP protokola koja su rezultirala ESMTP (eng. *Extended SMTP*) protokolom koji je danas najčešće u primjeni. Unatoč tome, ni ESMTP standard ne definira mehanizme autentikacije kod komunikacije ESMTP protokolom, a to ne čini niti jedan drugi standard, barem ne u potpunosti. Razni standardi definiraju razne implementacije autentikacijskih mehanizama za SMTP protokol, dok ih razni SMTP klijenti i poslužitelji selektivno podržavaju. Ovaj dokument uz opis SMTP protokola i njegovih ekstenzija daje prikaz najčešće korištenih i implementiranih metoda autentikacije za SMTP protokol.

2. Osnove SMTP protokola

Svrha SMTP protokola je pouzdan i efikasan prijenos poruka elektroničke pošte neovisno o komunikacijskom mediju koji se pritom koristi. SMTP za prijenos zahtijeva samo pouzdan, uređen podatkovni kanal, što u većini slučajeva podrazumijeva TCP komunikacijski kanal, iako je SMTP moguće koristiti i na drugim tipovima komunikacijskih kanala.
Važna funkcionalnost SMTP protokola je i njegova sposobnost prijenosa poruka elektroničke pošte između dvije ili više odvojenih mreža (eng. *SMTP mail relaying*) gdje se svaka mreža sastoji od poslužitelja dostupnih putem TCP protokola i Interneta te od poslužitelja dostupnih putem TCP/IP protokola koji su odvojeni od Interneta vatrozidima i dio su odvojene unutarnje mreže. Također, mreža se može satojati i od poslužitelja dostupnih putem nekog drugog protokola osim TCP/IP protokola koji su dio odvojenih lokalnih ili bežičnih mreža. Za prijenos poruka unutar mreže ili između mreža SMTP protokol koristi posredne poslužitelje koji su dostupni od strane mreža uključenih u komunikaciju te na taj način ostvaruje prijenos poruka od pošiljatelja do primatelja neovisno o tome koliko se različitih mreža nalazi između njih.

2.1. SMTP model

Funkcioniranje SMTP protokola može se prikazati sljedećom slikom.



Slika 1: Funkcionalnost SMTP protokola

Kad SMTP klijent dobije zahtjev za slanje poruke elektroničke pošte, on provjerava ime domene primatelja poruke i na osnovu imena domene određuje s kojim SMTP poslužiteljem mora komunicirati kako bi poslao poruku. Taj SMTP poslužitelj ne mora nužno biti odredišni poslužitelj već to može biti SMTP poslužitelj koji će nakon primitka preuzeti ulogu SMTP klijenta i proslijediti poruku do nekog drugog SMTP poslužitelja (eng. *relay*) ili to pak može biti tzv. izlazni (eng. *gateway*) SMTP poslužitelj koji će proslijediti poruku koristeći neki drugi protokol različit od SMTP protokola. To znači da se prijenos poruke može obaviti unutar samo jedne veze između SMTP pošiljatelja i SMTP primatelja ili može biti realiziran kao niz skokova s uspostavljanjem određenog broja komunikacijskih kanala između posredničkih klijenata/poslužitelja. U bilo kojem od navedenih slučaja dolazi do formalne predaje odgovornosti gdje SMTP poslužitelj prije nego zatvori komunikacijski kanal mora preuzeti obavezu dostavljanja/prosljeđivanja poruke ili izvijestiti klijenta u slučaju ako to nije u stanju učiniti.
Jednom kad se uspostavi komunikacijski kanal SMTP klijent inicira prijenos poruke elektroničke pošte koji uključuje izmjenu naredbi između klijenta i poslužitelja kojima se specificiraju pošiljatelj i primatelj. Nakon toga uzima se poruka s datotečnog sustava i obavlja se njen prijenos pri čemu su u sadržaj uključena i sva zaglavlja poruke. Kod slanja poruke prema više primatelja, SMTP protokol šalje samo jednu kopiju poruke svim primateljima koji se nalaze na istom poslužitelju (ili kojima se poruka prosljeđuje preko istog poslužitelja) radi uštede mrežnih resursa.
Poslužitelj na svaku naredbu mora odgovoriti pri čemu mogući dogovori mogu biti:

- naredba prihvaćena,
- očekuje se nastavak naredbe i
- došlo je do privremene ili trajne greške na poslužitelju.

Proces izmjenjivanja naredbi namjerno je organiziran po „1 na 1“ principu (1 naredba -> 1 odgovor), ali taj princip može biti izmijenjen ukoliko su između SMTP klijenta i poslužitelja uspostavljene za to

potrebne SMTP ekstenzije. Ekstenzijama se također može urediti i korištenje dodatnih SMTP naredbi ako su one podržane od obje strane.
Nakon uspješnog prijenosa poruke SMTP klijent može zatražiti prekid veze s poslužiteljem ili nastaviti s prijenosom drugih poruka elektroničke pošte. Isto tako može zatražiti i druge usluge kao što su na primjer verifikacija adresa elektroničke pošte ili dohvaćanje liste adresa elektroničke pošte s tog poslužitelja.
Kao što je opisano prijenos poruka elektroničke pošte se odvija između klijenta primatelja i poslužitelja primatelja poruke ako se i klijent i poslužitelj nalaze na istoj prijenosnoj mreži. Ukoliko ne postoji prijenosna mreža, tj. ne može se uspostaviti direktni prijenosni kanal, prijenos poruke se odvija preko jednog ili više relejnih (eng. *relay*) ili izlaznih (eng. *gateway*) SMTP poslužitelja. Informaciju o tome koji relejni ili izlazni SMTP poslužitelj treba kontaktirati, SMTP klijent dobiva korištenjem DNS (eng. *Domain name service*) poslužitelja i pripadnog DNS MX (eng. *Mail eXchanger*) zapisa pomoću kojeg se iz adrese primatelja dobiva podatak o nazivu SMTP poslužitelja koji će poruku proslijediti primatelju. Prije nego što se provjere DNS MX zapisi, SMTP klijent provjerava je li domena u adresi primatelja važeća i ukoliko jest tek onda pretražuje DNS MX zapise u potrazi za imenom odgovarajućeg SMTP poslužitelja. Ako se pokaže da je domena u adresi nevažeća SMTP klijent javlja pogrešku i ne prosljeđuje poruku elektroničke pošte.
Važno je napomenuti kako je SMTP protokol isključivo *push* protokol, tj. njime se može slati poruka, ali se ne može prozvati poslužitelj i s njega dohvatiti poruka (eng. *pull*). Kako bi to učinio, klijent mora podržavati POP3 ili IMAP protokol. Također, pošto je SMTP protokol isključivo tekstualni ASCII protokol, kao takav nije prikladan za slanje binarnih sadržaja i datoteka. U tu svrhu, razvijeni su MIME standardi koji specificiraju kodiranje binarnih podataka za prijenos SMTP protokolom.
Sendmail je poznat kao jedan od prvih poslužitelja elektroničke pošte koji podržava SMTP protokol, ali danas je broj takvih poslužitelja izuzetno velik (Postfix, Qmail, Microsoft Exchange Server, itd.).

2.2. SMTP procedure i naredbe

Procedura slanja poruka elektroničke pošte opisana u prethodnom poglavlju ostvaruje se izmjenom SMTP naredbi između klijenta i poslužitelja. Prvi korak u toj proceduri je uspostavljanje sesije, tj. otvaranje prijenosnog kanala između klijenta i poslužitelja. Vezu uspostavlja klijent i ako poslužitelj prihvati vezu, onda započinje razmjena SMTP naredbi točno utvrđenim poretkom. SMTP naredbe su tekstualni nizovi znakova koji završavaju sa znakom kraja retka <CRLF>, a u sebi mogu sadržavati parametre odvojene od same naredbe znakom <SP>. Osnovne SMTP naredbe opisane su u nastavku dokumenta.

2.2.1. EHLO / HELO naredba

Prva u nizu naredbi je EHLO naredba kojom se klijent identificira poslužitelju i kojom obavještava poslužitelj kako podržava ekstenzije SMTP protokola. Starije inačice klijenata koje ne podržavaju SMTP ekstenzije identificirat će se naredbom HELO. Sadržaj EHLO / HELO naredbe kojim se klijent identificira je naziv njegove domene. Poslužitelj će na primljenu EHLO naredbu odgovoriti EHLO odgovorom ili s porukom kako naredba nije prepoznata (eng. *command not recognized*) ako ne podržava SMTP ekstenzije. U tom slučaju klijent mora pokušati ponovnu identifikaciju slanjem HELO naredbe. Na primljenu HELO naredbu poslužitelj smije odgovoriti samo HELO odgovorom, a nikako ne EHLO odgovorom. Primjer izmjene EHLO naredbi prikazan je u nastavku (P – poslužitelj, K – klijent).

```
P: 220 smtp.primjer.hr ESMTP server ready          (prihvaćen kom. kanal)
K: EHLO jgm.primjer.hr                             (EHLO identifikacija)
P: 250-smtp.primjer.hr                             (EHLO odgovor)
```

2.2.2. MAIL naredba

Prvi korak u slanju poruke elektroničke pošte je MAIL naredba kojom se identificira pošiljatelj poruke. MAIL naredba sadrži informaciju o izvorišnoj adresi elektroničke pošte na koji se šalju poruke o eventualnim pogreškama. Poslužitelj može prihvatiti identifikaciju i poslati odgovor „*250 OK*“ ili prijaviti trajnu ili privremenu grešku porukama „*550*“ ili „*553*“. Greška je trajna ako poslužitelj ocijeni da je podatak o izvorišnoj adresi elektroničke pošte nevažeći, a privremena ako poslužitelj ocijeni kako

trenutno ne može prihvatiti takvu identifikaciju klijenta, ali je može prihvatiti ako klijent pokuša ponovo. Primjer izmjene MAIL naredbi prikazan je u nastavku.

```
K: EHLO pop3.primjer.hr
P: 250 OK
K: MAIL FROM: Ivo <ivo@primjer.hr>
P: 250 OK - mail from Ivo <ivo@primjer.hr>
```

2.2.3. RCPT naredba

Nakon MAIL naredbe slijedi jedna ili više RCPT naredbi kojima se specificira jedan ili više primatelja poruke. Argument RCPT naredbe je adresa elektroničke pošte primatelja poruke ili tzv. putanja prosljeđivanja (eng. *forward-path*). U slučaju kada je argument u redu i kada se poštovala prethodna procedura, odgovor na RCPT naredbu je „*250 OK*" nakon čega poslužitelj pohranjuje vrijednost putanje prosljeđivanja. Ako poslužitelj utvrdi da je vrijednost argumenta RCPT naredbe nevažeća adresa elektroničke pošte, onda odgovara s odgovorom „*550 No such user <putanja prosljeđivanja>*" koji označava da ne postoji navedeni korisnik. Prema osnovnoj inačici SMTP protokola, putanja prosljeđivanja ne mora nužno biti adresa elektroničke pošte već ona može sadržavati i odredišnu rutu prema primatelju. Međutim, ta se mogućnost danas ne koristi, tj. ako je argument RCPT naredbe odredišna ruta, SMTP poslužitelji jednostavno tu vrijednost ignoriraju i ne prosljeđuju takvu poruku. Također ako SMTP poslužitelj utvrdi da nije poštivan redoslijed SMTP naredbi u transakciji, tj. da prije RCPT naredbe nije bilo MAIL naredbe onda odgovara s porukom „*503 Bad sequence of commands*" koja označava nevažeći redoslijed naredbi. Primjer izmjene RCPT naredbi prikazan je u nastavku.

```
K: EHLO pop3.primjer.hr
P: 250 OK
K: MAIL FROM: Ivo <ivo@primjer.hr>
P: 250 OK - mail from Ivo ivo@primjer.hr
K: RCPT TO: Maja <maja@drugi_primjer.hr>
P: 250 OK - Recipient Maja <maja@drugi_primjer.hr>
```

2.2.4. DATA naredba

Nakon specificiranja pošiljatelja i primatelja poruke slijedi slanje sadržaja poruke pomoću naredbe DATA. Nakon primitka te naredbe poslužitelj odgovara među-odgovorom „*354*" i od tog trenutka interpretira svaki sljedeći primljeni redak kao sadržaj poruke dok ne primi indikator kraja poruke. Nakon toga, ako je cjelokupan sadržaj uspješno zaprimljen i pohranjen, SMTP poslužitelj odgovara s „*250 OK*" porukom i time potvrđuje prijem.
Kao indikator kraja poruke koristi se redak teksta u kojoj je samo znak točke "*<CRLF>.<CRLF>*". Kako bi se izbjegli nesporazumi i omogućilo korisniku slanje točki, koristi se procedura transparentnosti koja definira sljedeće:

- prije slanja retka teksta SMTP klijent provjerava prvi znak u retku – ako je to točka onda na početak tog retka ubacuje još jednu točku,
- SMTP poslužitelj provjerava primljene retke – ako redak sadrži samo točku onda to podrazumijeva kraj poruke, a ako je pak prvi znak retka točka i postoji još znakova u retku onda briše prvu točku.

Tekst ili sadržaj poruke može biti bilo koji od 128 ASCII znakova, međutim ako se u tekstu koriste neki drugi znakovi (npr. hrvatski znakovi), onda se prije slanja taj tekst mora transformirati u oblik pogodan za slanje putem SMTP protokola korištenjem reverzibilne transformacije kako bi se na strani primatelja sadržaj mogao rekonstruirati.
DATA naredba može imati i neuspješno izvršenje i to u 2 situacije:

- ako naredbi nije prethodila RCPT ili MAIL naredba, SMTP poslužitelj može vratiti ili „*503 Command out of sequence*" ili „*554 No valid recipients*" poruku te ako klijent primi takvu poruku od poslužitelja, on ne smije slati sadržaj poruke – sadržaj se smije slati tek nakon primitka poruke „*354*",
- ako je primljena poruka „*354*" na strani klijenta, DATA naredba može biti svejedno neuspješna ako nije bilo primatelja (neki SMTP poslužitelji verificiraju adresu primatelja tek

nakon primitka cijele poruke), ako su resursi poslužitelja postali nedostupni ili ako poslužitelj zbog svojih sigurnosnih postavki ne može prihvatiti poruku.

Primjer izmjene DATA poruke prikazan je u nastavku.

```
K: EHLO pop3.primjer.hr
P: 250 OK
K: MAIL FROM: Ivo <ivo@primjer.hr>
P: 250 OK - mail from Ivo ivo@primjer.hr
K: RCPT TO: Maja <maja@drugi_primjer.hr>
P: 250 OK - Recipient Maja <maja@drugi_primjer.hr>
K: DATA
K: 354 Send data. End with CRLF.CRLF
K: TO: Maja <maja@drugi_primjer.hr>
K: FROM: Ivo <ivo@primjer.hr>
K: DATE: Sat 22 Jun 2006 12:11:11 -0400
K: SUBJECT: Osnovna SMTP poruka
K: X-Info: Evaluation version at pop3.primjer.hr
K: Message-Id: <18143885500034@primjer.hr>
K:
K: Tekst poruke.
K:
K:.
K: 250 OK
```

2.2.5. Ostale osnovne SMTP naredbe

Minimalna implementacija SMTP protokola uz već navedene naredbe zahtijeva i implementaciju sljedećih naredbi:

- RSET – služi za prekid sesije i zahtijeva od poslužitelja pražnjenje svojih privremenih spremnika i tablica kojima pohranjuje podatke o primateljima poruka,
- VRFY – zahtjev poslužitelju da verificira argument naredbe kao adresu primatelja, tj. da provjeri da li odgovara korisničkom imenu ili adresi elektroničke pošte; naredba ne utječe na sadržaj spremnika koji pohranjuju argumente ostalih naredbi,
- NOOP – naredba koje ne pokreće nikakvu akciju i ne utječe na ostale naredbe nego samo zahtijeva od poslužitelj odgovor s „*250 OK*“ porukom,
- QUIT – naredba kojom se od poslužitelja zahtijeva slanje „*250 OK*“ poruke i zatvaranje komunikacijskog kanala; niti klijent niti poslužitelj ne smiju zatvoriti kanal bez te naredbe čak ni u slučaju pojave greške, itd…

Sve ostale naredbe su poželjne, ali ne i obavezne.

3. Ekstenzije SMTP protokola

Nakon početne specifikacije SMTP protokola definirane RFC821 standardom i početka korištenja, uočena je potreba za nadogradnjom protokola pa je definiran model za dodavanje ekstenzija protokola. Model definira način na koji klijent i poslužitelj mogu prepoznati ukoliko druga strana podržava ekstenzije SMTP protokola i koje su to ekstenzije.

Tako je definiran osnovni mehanizam ekstenzija koji podrazumijeva obaveznu podršku za naredbu EHLO čak i za poslužitelje i klijente koji ne podržavaju SMTP ekstenzije. Također, poslužitelji koji podržavaju SMTP ekstenzije moraju podržavati i komunikaciju SMTP protokolom bez ekstenzija, tj. u slučaju kada druga strana ne prepoznaje EHLO naredbu, oni moraju prijeći na upotrebu HELO naredbe.

Osim podrške za naredbu EHLO, SMTP model ekstenzija specificira i sljedeće:

- implementaciju registra SMTP ekstenzija,
- dodatne parametre SMTP, MAIL i RCPT naredbi i
- opcionalne zamjene za standardne SMTP naredbe, npr. za DATA prilikom prijenosa sadržaja koji nisu ASCII formata.

Za registar SMTP ekstenzija odgovorna je organizacija IANA (*Internet Assigned Numbers Authority*) koja verificira i odobrava registraciju novih službenih SMTP ekstenzija. Svaka ekstenzija vezana je uz određenu ključnu riječ pridruženu EHLO naredbi koja jednoznačno definira tu ekstenziju.

Definicija SMTP ekstenzije mora biti u obliku formalnog standarda i mora uključivati sljedeće:

- tekstualni naziv SMTP ekstenzije,
- EHLO ključnu riječ ekstenzije,
- opis sintakse i listu parametara koji se upotrebljavaju uz ekstenziju,
- listu dodatnih SMTP naredbi na koje ekstenzija utječe,
- listu dodatnih parametara koji se koriste uz MAIL i RCPT naredbe radi implementacije ekstenzije,
- opis koji definira utjecaj ekstenzije na ponašanje SMTP klijenta i poslužitelja, te
- specifikaciju potrebne promjene maksimalne dužine MAIL i/ili RCPT naredbi u odnosu na dužinu definiranu RFC2821 standardom.

Ostavljena je i mogućnost upotrebe neregistriranih SMTP ekstenzija – one moraju biti definirane EHLO ključnim riječima koje počinju slovom „X" i služe isključivo za lokalnu primjenu koja se dogovara bilateralnim sporazumom između poslužitelja i klijenta. Zbog ovog pravila službene registrirane SMTP ekstenzije ne smiju počinjati slovom „X".
Tablica koja slijedi daje pregled najčešće korištenih SMTP ekstenzija.

Ekstenzija	Dokumentacija	Opis
8BITMIME	RFC1652	Prijenos 8-bitnog teksta SMTP protokolom.
ATRN	RFC2645	Dohvaćanje poruka elektroničke pošte s poslužitelja.
AUTH	**RFC2554**	**Općenito sučelje za autentikaciju klijenta, podržava različite metode za autentikaciju, nije registriran od strane IANA organizacije.**
BINARYMIME	RFC3030	Prijenos binarnih podataka bilo kojeg formata (8BITMIME dozvoljava samo prijenos teksta koji nije 7-bitno kodiran); zahtijeva upotrebu CHUNKING ekstenzije.
CHECKPOINT	RFC 1845	Ekstenzija za implementaciju točki provjera i ponovni start.
CHUNKING	RFC3030	Za segmentiranje sadržaja poruke.
DSN	RFC1891	Javljanje statusa isporuke (eng. *Delivery Status Notification*).
ETRN	RFC1895	Slanje međusobno ovisnih poruka; ne preporučuje se korištenje ove ekstenzije; preporuka je koristiti POP ili slične protokole.
EXPN	RFC821	Otvaranje/prikazivanje liste adresa elektroničke pošte.
HELP	RFC821	Prikazivanje pomoćnih informacija u čitljivom obliku.
ONEX		Slanje samo jedne poruke; ne postoji službena dokumentacija.
PIPELINING	RFC2920	Vezanje naredbi.
SAML	RFC821	Slanje poruke elektroničke pošte terminalu; zastarjela nakon RFC2821.
SEND	RFC821	Slanje poruke elektroničke pošte udaljenom terminalu; zastarjela nakon RFC2821.
SIZE	RFC1870	Razmjena informacija o veličini poruka.
SOML	RFC821	Slanje poruke elektroničke terminalu; zastarjela nakon RFC2821.
STARTTLS	RFC3207	SMTP preko TLS sigurne veze.
TURN	RFC821	Inicijalna ekstenzija koja se više ne koristi - zastarjela nakon RFC2821.
XCLIENT		Simuliranje veze s klijentom.
XFORWARD		Simuliranje SMTP sesije – koristi se za popunjavanje informacija o primljenoj poruci elektroničke pošte, ali ne za kontrolu pristupa.
XVERP		Promjenjiva povratna ruta – korišteno od strane programa za kreiranje lista adresa elektroničke pošte.

Tablica 1: Najčešće korištene SMTP ekstenzije

4. SMTP autentikacija

SMTP autentikacija je ekstenzija SMTP protokola kojom se SMTP poslužiteljima omogućava implementiranje mehanizma autentikacije klijenta, obaviještavanje klijenta o potrebi za autentikacijom, a ukoliko klijent podržava autentikacijski mehanizam i provedbu autentikacije klijenta. Ekstenzija za SMTP autentikaciju definirana je RFC2554 standardom, ali samo u svojim osnovama jer se tim standardom definira samo naredba „AUTH" i njezina implementacija dok se ne specificiraju stvarni mehanizmi autentikacije koji se time mogu podržati.

4.1. AUTH naredba

AUTH naredbu koristi poslužitelj nakon otvaranja sesije s klijentom, tj. nakon što primi EHLO naredbu od klijenta poslužitelj odgovara odgovorom „*250 OK*". Nakon toga poslužitelj obavještava klijenta da podržava ekstenziju za autentikaciju te u argumentu AUTH naredbe specificira i koji mehanizam autentikacije podržava (jedan ili više – odvojeni razmakom). Argument AUTH naredbe također može specificirati i sigurnosni protokol (eng. SASL – *Simple Authentication and Security Layer*) koji će se koristiti za zaštitu komunikacije nakon uspješne provedbe autentikacije.
Odgovor na poslužiteljevu obavijest o potrebi za autentikacijom je također AUTH naredba kojom klijent obavještava poslužitelja kako podržava jednu od navedenih mehanizama autentikacije navođenjem tog mehanizma kao argumenta AUTH naredbe. Ako klijent ne podržava ni jedan od traženih mehanizama autentikacije, tada on šalje odgovor „*504*" i time sesija završava.
Poslužitelj nakon dogovora o metodi autentikacije klijentu šalje među-odgovor „*334*" te odmah nakon toga počinje proces autentikacije koji se sastoji u izmjeni poslužiteljevih izazova (eng. *challenge*) i klijentovih odgovora. Izazovi i odgovori su obično BASE64 kodirani nizovi znakova kodirani prema već odabranom mehanizmu autentikacije. Ako odgovori klijenta odgovaraju traženom izazovu i autentikacija je uspješno provedena, poslužitelj odgovara odgovorom „*235*" i sesija se nastavlja dalje uobičajenim tokom kao što je definirano SMTP protokolom.
Ako poslužitelj nakon slanja izazova dobije odgovor koji ne može dekodirati, tada on šalje odgovor „*501*", a ako ga može dekodirati pri čemu isti on ne odgovara poslanom izazovu, tada poslužitelj šalje odgovor „*535*" te neki od mogućih kodova greške ako su prikladni. Ako pak klijent želi prekinuti proces autentikacije, kao odgovor na izazov šalje redak koji sadrži samo znak „ *" na što poslužitelj odgovara s „*501*" i odbija autentikaciju.
Ako je autentikacijom dogovoren i sigurnosni protokol za daljnju komunikaciju, on se počinje koristiti odmah nakon uspješno završene autentikacije, tj. za poslužitelja odmah nakon kraja zadnjeg retka odgovora „*235*", a za klijenta nakon primitka tog odgovora. Nakon toga uspostavlja se dogovoreni sigurnosni protokol SASL (gdje se specificira SMTP kao usluga koju će prenositi) i SMTP komunikacija počinje ispočetka tako što klijent šalje EHLO naredbu.
Nakon uspješno provedene autentikacije ne smije se više pojaviti ni jedna AUTH naredba za vrijeme iste sesije – ako se i pojavi, poslužitelj/klijent je mora odbiti odgovorom „*503*". Primjer procesa autentikacije SMTP protokolom prikazan je u nastavku.

```
P: 220 smtp.primjer.hr ESMTP server ready
K: EHLO jgm.primjer.hr
P: 250-smtp.primjer.hr
P: 250 AUTH CRAM-MD5 DIGEST-MD5
K: AUTH FOOBAR
P: 504 Nepoznata metoda autentikacije
K: AUTH CRAM-MD5
P: 334
PENCeUxFREJoU0NnbmhNWitOMjNGNndAZWx3b29kLmlubm9zb2Z0LmNvbT4=
K: ZnJlZCA5ZTk1YWVlMDljNDBhZjJiODRhMGMyYjNiYmFlNzg2ZQ==
P: 235 Autentikacija uspješna
```

U gore navedenom primjeru vidljivo je korištenje CRAM MD-5 mehanizama koji je kao i DIGEST MD-5 naveden kao jedan od mogućih mehanizama koji se spominju u RFC2554 standardu. Međutim, taj standard ne definira nijedan od ta dva navedena mehanizma već samo specificira da SMTP mora podržati izmjenu izazova i odgovora koji mogu biti neodređene dužine, tj. koja god dužina bila

potrebna za implementaciju određenog mehanizma, SMTP protokol je mora podržati. Ove dvije metode kao i druge često korištene biti će objašnjene u nastavku dokumenta.

5. Metode SMTP autentikacije

Iako nisu standardizirane nijednom ekstenzijom SMTP protokola, najčešće korištene metode SMTP autentikacije su sljedeće:

- AUTH LOGIN,
- AUTH PLAIN,
- CRAM-MD5,
- Kerberos V4,
- GSSAPI i
- S/Key.

Kratak opis svake od ovih metoda slijedi u nastavku dokumenta.

5.1. LOGIN autentikacija

Ova jednostavna metoda autentikacije zasniva se na korisničkom imenu i zaporki i BASE64 kodiranju niza znakova. Kod ovakve autentikacije nakon što poslužitelj i klijent dogovore LOGIN kao metodu autentikacije, poslužitelj šalje klijentu zahtjev za korisničkim imenom, na što klijent odgovara stvarnim korisničkim imenom. Nakon toga poslužitelj zahtijeva i zaporku i nakon što je dobije od klijenta te je verificira, autentikacija je uspješno završena. Treba napomenuti da su zahtjevi i odgovori za korisničkim imenom i zaporkom BAE64 kodirani tako da se autentikacijski podaci ne šalju kao čitljivi nizovi znakova. Primjer LOGIN autentikacije prikazan je u nastavku.

```
P: 220 esmtp.primjer.hr ESMTP
K: ehlo client.primjer.hr
P: 250-esmtp.primjer.hr
P: 250-PIPELINING
P: 250-8BITMIME
P: 250-SIZE 255555555
P: 250 AUTH LOGIN PLAIN CRAM-MD5
K: auth login
P: 334 VXNlcm5hbWU6
K: a29yaXNuaWs=
P: 334 UGFzc3dvcmQ6
K: emFwb3JrYQ==
P: 535 Autentikacija neuspješna
```

U gore navedenom primjeru od ponuđenih metoda autentikacije (LOGIN, PLAIN, CRAM-MD5) klijent odabire LOGIN metodu nakon čega poslužitelj šalje niz znakova „Username:“ koji kad je BASE64 kodiran izgleda kao „`VXNlcm5hbWU6`“. Klijent na to odgovara slanjem korisničkog imena (korisnik) također kodiranog u BASE64 formatu „`a29yaXNuaWs=`“. Poslužitelj nakon toga zahtijeva i zaporku za to korisničko ime slanjem niza znakova „Password:“ (BASE64 – „`UGFzc3dvcmQ6`“) na što klijent odgovara slanjem vrijednosti zaporke u BASE64 formatu „`emFwb3JrYQ==`“. U ovom slučaju zaporka nije ispravna i poslužitelj obavještava klijenta kako autentikacija nije bila uspješna.

5.1.1. BASE64 kodiranje

BASE64 kodiranje je predviđeno za kodiranje slučajnog niza okteta u obliku koji nije nužno čitljiv. Algoritmi za kodiranje i dekodiranje su jednostavni, a posljedica je da je kodirani niz znakova gotovo uvijek 33% veći nego originalni niz znakova.

Za zapis kodiranog niza znakova se koristi skup od 65 znakova koji je podskup US-ASCII skupa znakova koji omogućava da 6 bitova bude zapisano kao jedan znak ($2^6 = 64$). Dodatni 65-ti znak je znak „=“ koji označava specijalnu procesnu funkciju.

Proces kodiranja uzima 24-bitne grupe ulaznih bitova i kodira ih kao izlazni niz od 4 kodirana znaka. To znači da se na ulazu iz originalnog niza znakova uzimaju 3 8-bitne grupe (3 znaka zapisana

pomoću 8 bita) i spajaju u 24-bitni niz koji se tretira kao 4 spojene 6-bitne grupe. Svaka 6-bitna grupa se zatim kodira kao jedan znak BASE64 abecede prema prikazanoj tablici, a dobiveni izlazni niz znakova slaže se u retke maksimalne dužine 76 znakova. Prilikom kodiranja pretpostavlja se da su ulazni bitovi poredani tako da im je prvi bit najteži bit (eng. *most significant bit*), a osmi bit najlakši (eng. *least significant bit*). Ako se dogodi da je na kraju niza potrebno kodirati niz koji ima manje od 24 bita vrši se dodavanje bitova do 24-og prema specijalnoj proceduri definiranoj standardom.

Broj	Znak	Broj	Znak	Broj	Znak	Broj	Znak
0	A	17	R	34	i	51	z
1	B	18	S	35	j	52	0
2	C	19	T	36	k	53	1
3	D	20	U	37	l	54	2
4	E	21	V	38	m	55	3
5	F	22	W	39	n	56	4
6	G	23	X	40	o	57	5
7	H	24	Y	41	p	58	6
8	I	25	Z	42	q	59	7
9	J	26	a	43	r	60	8
10	K	27	b	44	s	61	9
11	L	28	c	45	t	62	+
12	M	29	d	46	u	63	/
13	N	30	e	47	v	pad	=
14	O	31	f	48	w		
15	P	32	g	49	x		
16	Q	33	h	50	y		

Tablica 2: BASE64 abeceda

Kao što je vidljivo iz abecede u njoj nema specijalnih znakova koji se koriste u SMTP protokolu (npr. „.“,CR, LF). Ako se prilikom dekodiranja naiđe na neki od znakova koji nije definiran BASE64 abecedom to je vrlo vjerojatno znak pogreške u prijenosu, te je u tom slučaju potrebno poslati poruku o grešci kao odgovor.

5.2. PLAIN autentikacija

PLAIN autentikacija gotovo je jednaka prethodno opisanoj LOGIN autentikaciji. Razlika je što se PLAIN autentikacijom korisnički podaci šalju jednom naredbom, a poslužitelj ne šalje „Username:“ i „Password:“ nizove znakova. Cijeli mehanizam autentikacije se sastoji od jednog niza znakova kodiranog po BASE64 tablici koji klijent šalje poslužitelju. Taj niz znakova sastoji se od autorizacijskog imena, autentikacijskog korisničkog imena i zaporke pri čemu su ti podaci međusobno odvojeni s NULL znakom ('\0'). Na sljedećem primjeru je niz poslan od klijenta poslužitelju kao dio

procesa PLAIN autentikacije. U primjeru je niz „test\0test\0testpass“ kodiran u BASE64 niz „dGVzdAB0ZXN0AHRlc3RwYXNz“.

```
P: 220 esmtp.primjer.hr ESMTP
K: ehlo client.primjer.hr
P: 250-esmtp.primjer.hr
P: 250-PIPELINING
P: 250-8BITMIME
P: 250-SIZE 255555555
P: 250 AUTH LOGIN PLAIN CRAM-MD5
K: AUTH PLAIN
P: 334
K: dGVzdAB0ZXN0AHRlc3RwYXNz
P: 235 Autentikacija uspješna
```

5.3. Anonimna autentikacija

Anonimna autentikacija se sastoji od jedne poruke od klijenta prema poslužitelju u kojoj klijent šalje opcionalnu informaciju o svom identitetu. Ta informacija može biti proizvoljni niz znakova koji ne sadrži znak „@“, a pošto je slanje opcionalno klijent ju može i ne poslati pa ovo stoga i nije prava autentikacija iako klijent može poslati određene identifikacijske podatke. Primjer ovakve autentikacije dan je u nastavku, a u tom primjeru korisnik samo šalje poruku „Ivo“ koja je u BASE64 formatu jednaka „SXZv“.

```
P: 220 esmtp.primjer.hr ESMTP
K: ehlo client.primjer.hr
P: 250-esmtp.primjer.hr
P: 250-PIPELINING
P: 250-8BITMIME
P: 250-SIZE 255555555
P: 250 AUTH ANONYMOUS
K: auth ANONYMOUS SXZv
P: 235 OK
```

5.4. CRAM-MD5 autentikacija

CRAM-MD5 metoda autentikacije se zasniva na izmjeni izazova (eng. *challenge*) i odgovora (eng. *response*) između klijenta i poslužitelja pa otuda dolazi i skraćenica CRAM (eng. *Challenge Response Authentication Mechanism*).
Ako klijent i poslužitelj dogovore primjenu ove metode autentikacije, poslužitelj šalje klijentu BASE64 kodirani pseudo slučajni niz znakova koji se sastoji od slučajnog niza znakova, jedinstvene identifikacije poslužitelja koja se sastoji od oznake vremena i punog naziva poslužitelja (format identifikacije poslužitelja definiran je RFC822 standardom).
Klijent na to odgovara BASE64 kodiranim nizom znakova koji se sastoji od korisničkog imena, razmaka i tzv. MD5 sume (eng. *MD5 digest*). MD5 suma se izračunava pomoću KEYED-MD5 algoritma (definiranog RFC1321 standardom), a kao ključ algoritma se koristi zajednička tajna (eng. *shared secret*) između poslužitelja i klijenta. Zajednička tajna je niz znakova koji je poznat samo klijentu i poslužitelju prije procesa autentikacije.

```
P: 220 smtp.primjer.hr ESMTP server ready
K: EHLO jgm.primjer.hr
P: 250-smtp.primjer.hr
P: 250 AUTH CRAM-MD5 DIGEST-MD5
K: AUTH CRAM-MD5
P: 334
P: + PDE4OTYuNjk3MTcwOTUyQHBvc3RvZmZpY2UucmVzdG9uLm1jaS5uZXQ+
K: dGltIGI5MTNhNjAyYzdlZGE3YTQ5NWI0ZTZlNzMzNGQzODkw
P: A0001 OK CRAM Autentikacija uspješna
```

U ovom primjeru poslužitelj kao izazov šalje BASE64 kodiran niz znakova „*<1896.697170952@postoffice.reston.mci.net>*“ koji u BASE64 obliku izgleda kao sljedeći niz znakova:

```
PDE4OTYuNjk3MTcwOTUyQHBvc3RvZmZpY2UucmVzdG9uLm1jaS5uZXQ+
```

Klijent dekodira primljeni niz znakova te pomoću njega i zajedničke tajne, koja je u ovom primjeru niz znakova „*tanstaaftanstaaf*“, izračunava MD5 sumu prema sljedećoj formuli:

MD5 suma = MD5[(tanstaaftanstaaf XOR opad), MD5((tanstaaftanstaaf XOR ipad), <1896.697170952@postoffice.reston.mci.net>)]

gdje su „*opad*“ i „*ipad*“ vrijednosti definirane KEYED-MD5 algoritmom. Ovim izračunom dobiva se sljedeći niz znakova u heksadecimalnom formatu:

```
b913a602c7eda7a495b4e6e7334d3890
```

Ispred dobivenog niza znakova dodaje se korisničko ime:

```
tim b913a602c7eda7a495b4e6e7334d3890
```

Dobiveni niz znakova kodira se u BASE64 format i time se dobiva konačan niz znakova koji se šalje poslužitelju kao odgovor na izazov:

```
dGltIGI5MTNhNjAyYzdlZGE3YTQ5NWI0ZTZlNzMzNGQzODkw
```

Treba napomenuti da se u ovom procesu zajednička tajna, ako je kraća od 64 okteta (eng. *byte*), produžuje nulama do duljine od 64 okteta, a ako je duža od 64 okteta onda u daljnju kalkulaciju ulazi samo MD5 suma od 16 okteta dobivena od zajedničke tajne.
Kad poslužitelj primi odgovor na izazov verificira ga reverznim izračunom korištenjem zajedničke tajne koja je i njemu poznata i ako je verifikacija uspješna odobrava pristup SMTP uslugama.

5.5. Kerberos V4 autentikacija

Kerberos V4 autentikacijski mehanizam također se bazira na razmjeni izazova i odgovora između klijenta i poslužitelja. Prvi izazov šalje poslužitelj i on sadrži 32-bitni broj u tzv. „*network byte order*“ formatu (uobičajeni format prijenosa podataka TCP/IP protokolom gdje je prvi oktet najznačajniji). Klijent odgovara porukom koja sadrži sljedeće podatke:

- Kerberos karticu (eng. *ticket*) za pristup Kerberos autentikacijskom mehanizmu koja se dodjeljuje klijentima od strane Kerberos centra za distribuciju kartica (eng. KDC – *Key Distribution Center*) te stoga klijent posjeduje karticu prije same autentikacije.
- Autentikator klijenta - identifikator klijenta kojem je dodijeljena Kerberos kartica. Format je specificiran sljedećim nizom "*service.hostname@realm*" gdje je:
  - *Service* – naziv usluge za koju se provodi autentikacija,
  - *Hostname* – prvi dio naziva poslužitelja na kojem se nalazi klijent; sve malim slovima,
  - *Realm* – domena poslužitelja na kojem se nalazi klijent.
- Kontrolna suma autentikatora - kriptirani niz znakova kreiran na osnovu autentikatora koji sadrži i 32 bitni broj primljen od poslužitelja.

Nakon provjere i verifikacije primljene kartice i autentikatora, poslužitelj verificira i kontrolnu sumu autentikatora i provjerava ukoliko ona sadrži poslani 32-bitni broj. Ako je sve uspješno verificirano, poslužitelj dodaje „1“ kontrolnoj sumi i kreira 8 okteta podataka sljedećeg sadržaja:

- prva 4 okteta sadrže uvećanu kontrolnu sumu autentikatora,

- peti oktet sadrži bit masku kojom se specificiraju sigurnosni protokoli koje poslužitelj podržava,
- 6, 7 i 8 oktet sadrže podatke o maksimalnoj veličini privremenog spremnika za enkripciju koju poslužitelj može rezervirati.

Poslužitelj kriptira navedenih 8 okteta DES ECB (eng. *Data Encryption Standard Electronic Code Book*) metodom korištenjem ključa sesije i šalje te podatke kao drugi izazov klijentu. Klijent provjerava prva 4 okteta primljenih podataka i ako odgovaraju prethodno poslanoj kontrolnoj sumi uvećanoj za 1, smatra poslužitelj autenticiranim. Nakon toga klijent kreira poruku sljedećeg sadržaja:

- prva 4 okteta sadrže originalnu kontrolnu sumu,
- peti oktet sadrži bit masku kojom se specificira sigurnosni protokol,
- 6, 7 i 8 oktet sadrže podatak o maksimalnoj veličini privremenog spremnika za enkripciju koju klijent može rezervirati,
- ostali okteti sadrže autorizacijski identitet.

Tako kreiranu poruku potrebno je proširiti oktetima popunjenim s nulama kako bi ukupna dužina poruke bila višekratnik od 8 i enkriptirati DES PCBC (eng. *Propagating Cipher-Block Chaining*) metodom i korištenjem ključa sesije te takvu poslati poslužitelju. Razlika između ECB i PCBC metoda je u tome što ECB kriptira samo čisti tekst, a PCBC kriptira kombinaciju čistog teksta i prethodno kriptiranog teksta. Poslužitelj dekriptira primljenu poruku, verificira kontrolnu sumu i provjerava ukoliko je dani autorizacijski identitet vezan uz prethodno danu Kerberos karticu. Ako jest, onda je klijent autenticiran i proces autentikacije je završen.

Sigurnosni protokol koji se nakon toga koristi specificiran je bit maskom sa sljedećim značenjima:

- 1 – ne koristi se sigurnosni protokol,
- 2 – zaštita integriteta (*krb_mk_safe*) i
- 4 – zaštita privatnosti (*krb_mk_priv*).

Ostale bit maske se ne koriste. Primjer Kerberos V4 autenikacije prikazan je u nastavku.

```
P: * OK IMAP4 Server
K: A001 AUTHENTICATE KERBEROS_V4
P: + AmFYig==
K: BAcAQU5EUkVXLkNNVS5FRFUAOCAsho84kLN3/IJmrMG+25a4DT
   +nZImJjnTNHJUtxAA+o0KPKfHEcAFs9a3CL5Oebe/ydHJUwYFd
   WwuQ1MWiy6IesKvjL5rL9WjXUb9MwT9bpObYLGOKi1Qh
P: + or//EoAADZI=
K: DiAF5A4gA+oOIALuBkAAmw==
P: A001 OK Kerberos V4 autentikacija uspješna
```

5.6. GSSAPI autentikacija

GSSAPI (eng. *Generic Security Services Application Programmable Interface*) mehanizam autentikacije predstavlja metodu za implementaciju mehanizma autentikacije, tj. daje standardizirano sučelje (eng. API – *Application Programmable Interface*) za implementaciju autentikacijskog mehanizma,

GSSAPI sam po sebi ne osigurava sigurnost komunikacije već sigurnost daje implementacija GSSAPI-a koju pružaju proizvođači sigurnosnih softvera i to najčešće u obliku softverskih biblioteka. Takve biblioteke posjeduju GSSAPI kompatibilno sučelje koje neka aplikacija, u ovom slučaju SMTP klijentska/poslužiteljska aplikacija, može iskoristiti za implementaciju autentikacije pomoću standardiziranih poziva funkcija. U takvoj situaciji ako se želi promijeniti autentikacijski mehanizam, aplikaciju ne treba mijenjati već je potrebno samo koristiti neki drugi autentikacijski mehanizam koji također podržava GSSAPI sučelje.

Princip GSSAPI autentikacije može se objasniti opisom izmjene poruka između poslužitelja i klijenta, te poziva pojedinih funkcija GSSAPI sučelja na strani klijenta, odnosno poslužitelja. Početak autentikacije je prazna poruka poslužitelja prema klijentu. Klijent nakon toga poziva funkciju *GSS_Init_sec_context* s argumentom „0“ i *targ_name* parametrom vrijednosti „X“ gdje je X prethodno dobiven pozivom funkcije *GSS_Import_Name* s argumentom „*SERVICE:imap@hostname*“ pri čemu „*hostname*“ predstavlja puni naziv poslužitelja na kojem se klijent nalazi.

GSS_Init_sec_context nakon ovakvog poziva vraća izlazni kod GSS_COMPLETE i generira izlazni niz znakova koji klijent potom šalje poslužitelju. Ako *GSS_Init_sec_context* vrati izlazni kod

GSS_COMPLETE i generira prazan niz znakova onda klijent šalje praznu poruku poslužitelju. Ako pak *GSS_Init_sec_context* nakon ovakvog poziva vraća izlazni kod GSS_CONTINUE_NEEDED onda klijent čeka da mu poslužitelj pošalje dio koda izazova i taj kod prosljeđuje funkciji *GSS_Init_sec_context*.
Nakon slanja inicijalnog niza znakova klijent očekuje poruku od poslužitelja u kojoj se nalazi kod izazova. Za interpretaciju primljenog koda izazova klijent koristi funkciju *GSS_Unseal* koja iz koda izvlači sljedeće podatke:

- prvi oktet – bit maska koja specificira sigurnosni mehanizam koji poslužitelj podržava,
- 2 – 4 oktet – maksimalna veličina poruke koju poslužitelj može primiti.

Nakon uspješne interpretacije koda klijent koristi funkciju *GSS_Seal* kojom generira izlaznu poruku koja sadrži sljedeće podatke:

- prvi oktet – bit maska koja specificira sigurnosni mehanizam koji klijent podržava,
- 2 – 4 oktet – maksimalna veličina poruke koju klijent može primiti,
- ostali okteti – korisničko ime.

Takvu poruku klijent šalje poslužitelju koji kad primi tu poruku od klijenta, koristi *GSS_Unseal* funkciju kako bi interpretirao njen sadržaj, verificirao primljene podatke, provjerio ukoliko za dano korisničko ime postoji definiran pristup traženoj usluzi te prihvatio autentikaciju klijenta.
Implementacija navedenih GSSAPI funkcija je proizvoljna, a u velikom broju slučajeva koristi se za implementaciju Kerberos autentikacijskog mehanizma.

5.7. S/Key autentikacija

S/Key autentikacijski mehanizam zasniva se na autentikaciji jednokratnom zaporkom. U tom slučaju poslužitelj na osnovu inicijalnog tajnog niza znakova i kriptografske funkcije generira N zaporki koje su međusobno povezane tako da jedna služi kao osnova za kreiranje sljedeće. Zaporke se nakon generiranja daju korisniku na korištenje, a inicijalni niz znakova i sve osim zadnje generirane zaporke se brišu s poslužitelja.
Klijent se autenticira slanjem predzadnje zaporke s liste koju je dobio s tim što u poruci šalje zaporku i njen redni broj, a sve kriptirano pomoću niza znakova koji je primio u inicijalnoj poruci od poslužitelja. Poslužitelj kad primi zaporku obavlja njenu verifikaciju upotrebom istog algoritma koji je korišten za generiranje zaporki te za rezultat dobiva zadnju zaporku, tj. onu koja je jedina ostala pohranjena na poslužitelju. Nakon toga poslužitelj sprema primljenu zaporku kao referentnu koju će koristiti za sljedeći proces autentikacije i tako nastavlja dok se ne potroše sve generirane zaporke.
Pošto se svaka zaporka koristi samo jednom, ako netko uspije presresti poruku sa zaporkom i dekriptirati je, ipak ne može neovlašteno dobiti važne podatke jer tu zaporku ne može ponovo iskoristiti, a samo na osnovu nje je gotovo nemoguće generirati sljedeću zaporku u nizu.

5.8. NTLM autentikacija

NTLM je autentikacijski protokol koji se koristi u raznim implementacijama Microsoft-ovih mrežnih protokola. Originalno se NTLM koristio za autentikaciju i sigurnost kod poziva udaljenih procedura (eng. RPC – *Remote Procedure Call*), a danas se koristi u Microsoft-ovim sustavima kao integrirani mehanizam za autentikaciju na jednom mjestu (eng. *Single Sign On*). Sama skraćenica dolazi od prvobitne namjene protokola kao sigurnosnog mrežnog protokola za Windows NT operacijske sustave (eng. *NT LAN Manager*).
NTLM je baziran na izmjeni poruka izazova i odgovora između poslužitelja i klijenta, ali kod NTLM autentikacije klijent dokazuje svoj identitet bez slanja zaporke poslužitelju. NTLM autentikacija se sastoji od 3 poruke koje su označene tipovima:

- tip 1 (dogovor),
- tip 2 (izazov) i
- tip3 (autentikacija).

Princip rada je sljedeći:

1. Klijent šalje poruku tipa 1 koja sadrži skup zastavica kojima klijent daje informaciju o tome koje usluge i opcije podržava, a koje traži od poslužitelja.

2. Poslužitelj odgovara porukom tipa 2 koja sadrži sličan skup zastavica kojima poslužitelj oglašava koje usluge i opcije podržava, a koje traži od klijenta čime se efektivno dogovaraju parametri autentikacije. Osim toga poruka sadrži i 8 oktalni slučajno generirani izazov.
3. Klijent korištenjem dobivenog izazova i identifikacijskih podataka o korisniku izračunava odgovor. Metoda izračunavanja ovisi o dogovorenim parametrima autentikacije, ali se u većini slučajeva za izračun koristi MD4/MD5 algoritam i DES metoda enkripcije za zaštitu odgovora. Konačni rezultat klijent šalje poslužitelju u poruci tipa 3 koju poslužitelj dekriptira i verificira te na osnovu toga autenticira klijenta/korisnika.

Ključan dio procedure je izračun odgovora kojim klijent, iako ne šalje zaporku, uspijeva dokazati kako zna zaporku. Klijent to može učiniti na više načina, tj. slanjem sljedećih tipova odgovora:

- LM odgovor – originalni tip odgovora; šalje ga većina klijenata,
- NTLM odgovor – šalju ga NT bazirani klijenti (Windows 2000, Windows XP),
- NTLMv2 odgovor – noviji tip odgovora koji je uveden pojavom Windows NT Service Pack 4 nadogradnje; zamjenjuje NTLM odgovor kod sustava koji podržavaju NTLMv2 i
- NTLM2 odgovor – odgovor koji se koristi u situaciji kada se dogovara NTLM2 sigurnosna sesija bez NTLMv2 autentikacije čime se mijenja semantika LM i NTLM odgovora.

U osnovi svi tipovi odgovora koriste zaporku ili njene dijelove koje na određeni način kombiniraju kako bi kreirali ključeve koji se onda koriste za DES enkripciju dobivenog izazova. Zaporka se nikad ne prenosi pa ju je nemoguće otkriti praćenjem komunikacije između klijenta i poslužitelja.

6. Sigurnost SMTP protokola

Sigurnost SMTP protokola izravno je povezana s prethodno opisanim metodama autentikacije korisnika. Naime, glavni nedostatak izvornog oblika SMTP protokola nalazi se u korištenju nekriptiranog teksta za uspostavljanje konekcije i prijenos poruka elektroničke pošte. Stoga se zbog sve većeg broja zlonamjernih korisnika prethodno opisane metode autentikacije sve češće koriste u radnim okruženjima.

SMTP autentikacija je mehanizam za autentikaciju SMTP klijenata kojim se može ograničiti neovlaštena upotreba SMTP protokola na osnovu identiteta klijenta. Ipak, autentikacijom se ne može garantirati autentičnost informacija koje se šalju u samoj poruci pa čak ni onih iz zaglavlja (pošiljatelj, primatelj), tako da je lažiranje poruka elektroničke pošte moguće unatoč provedenoj autentikaciji. Zbog toga i postoji problem tzv. „spam“ poruka elektroničke pošte. Također, ako je samo autenticiranim korisnicima dozvoljeno prosljeđivanje poruka elektroničke pošte i ako zlonamjerni korisnik otkrije zaporku nekog važećeg korisnika, pristup na poslužitelj mu je otvoren i on ga može koristiti za prosljeđivanje i slanje velikih količina lažnih poruka jer se njihov sadržaj, kao ni pošiljatelj ni primatelj ne mogu verificirati. Stoga je nužno korištenje naprednijih metoda SMTP autentikacije kojima se bolje štite korisnikov identitet i zaporka te se otežava neovlašteno autoriziranje korisnika što bitno povećava razinu sigurnosti sustava.

SMTP autentikacija također daje opciju SMTP klijentu da potvrdi kako je neki korisnik već autenticiran te jednostavno traži od poslužitelja prosljeđivanje njegove poruke elektroničke pošte, čime se u biti traži od poslužitelja da vjeruje klijentu. Takvo ponašanje je posljedica povijesnog razvoja SMTP protokola koji je nastao u vremenima kada su Internet i poruke elektroničke pošte koristili rijetki korisnici koji su većinom bili stručnjaci i koji nisu zlorabili tu novu uslugu. Međutim proširenjem i popularizacijom Interneta i računalnih tehnologija, korisnici postaju svi pa tako i zlonamjerni korisnici te se stoga pristupilo implementaciji autentikacijskih mehanizama.

Zbog svega navedenog, osim implementacije jedne od metoda SMTP autentikacije, administratorima SMTP poslužitelja se preporučuju i sljedeće mjere sigurnosti:

- provjera IP adrese klijenta koji se pokušava povezati pomoću DNS poslužitelja ili odbijanje veze ako klijent nije na listi autoriziranih klijenata,
- onemogućavanje pojedinih SMTP naredbi i funkcija kao što su prosljeđivanje poruka za neautorizirane klijente,
- ograničenje upotrebe EXPN naredbe kako bi se onemogućilo neautoriziranim korisnicima uvid u pojedinačne adrese skrivene iza grupe adresa,
- provjera podataka iz zaglavlja poruke prije prihvaćanja slanja poruke – uključuje provjeru adrese pošiljatelja i primatelja – slanje se odbija ako je bilo koja nevažeća,

- ograničavanje veličine poruka elektroničke pošte ili broja poruka koje mogu biti poslane u nekom vremenskom periodu,
- zapisivanje i praćenje svih pristupa poslužitelju kako bi se otkrile zlouporabe, itd…

Iako ni ove mjere ne garantiraju potpunu zaštitu od neautorizirane upotrebe, trebale bi je bitno smanjiti dok u širu upotrebu ne uđe neki protokol koji će moći jeftino i efikasno zamijeniti SMTP protokol.

7. Zaključak

SMTP protokol je veoma popularan protokol za razmjenu poruka elektroničke pošte čemu doprinose i niski troškovi implementacije i administracije kao i jednostavna i pouzdana shema prijenosa poruka elektroničke pošte preko većeg broja poslužitelja. Stoga ga većina proizvođača ugrađuje u svoje platforme. Upravo ta jednostavna shema je i glavni nedostatak SMTP protokola jer ne podržava sve potrebne elemente pa je potrebno koristiti ekstenzije, a unutar te jednostavnosti nalaze se i sigurnosni problemi. Stoga se sve važniji aspekti stavljaju upravo na sigurnost SMTP protokola pri čemu važnu ulogu ima upravo SMTP autentikacija.

Trenutno su raspoloživi različiti oblici SMTP autentikacijskih metoda koji otežavaju ili u potpunosti onemogućavaju zlonamjerne napadače u njihovim aktivnostima. Bez tih autentikacijskih metoda i bez primjene ostalih sigurnosnih kontrolnih metoda, poslužitelji elektroničke pošte bi brzo postali metom različitih oglašivača koji nastoje putem elektroničke pošte slati što veće količine svojih oglasnih *spam* poruka, ali i zlonamjernih napadača koji se mogu neovlašteno predstavljati kao određeni korisnici ili onemogućavati ispravan rad poslužitelja.

8. Reference

[1] RFC2821 - Simple Mail Transfer Protocol, http://www.ietf.org/rfc/rfc2821.txt
[2] RFC2554 - SMTP Service Extension for Authentication, http://www.ietf.org/rfc/rfc2554.txt
[3] RFC2045 - Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies, http://www.ietf.org/rfc/rfc2045.txt
[4] RFC2222 - Simple Authentication and Security Layer (SASL), http://www.ietf.org/rfc/rfc2222.txt
[5] RFC2245 – Anonymous SASL Mechanism, http://www.ietf.org/rfc/rfc2245.txt
[6] TCP/IP Guide – SMTP Security Issues, http://www.tcpipguide.com/free/t_SMTPSecurityIssues.htm, svibanj 2006.
[7] NTLM Authentication Protocol - http://curl.haxx.se/rfc/ntlm.html#ntlmSmtpAuthentication, svibanj 2006.
[8] SMTP Authentication, http://www.fehcom.de/qmail/smtpauth.html, svibanj 2006.