HRVATSKA AKADEMSKA I ISTRAŽIVAČKA MREŽA
CROATIAN ACADEMIC AND RESEARCH NETWORK

Nedostaci CHAP autentikacije

CCERT-PUBDOC-2002-03-01

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

CHAP se najčešće koristi u sprezi sa PPP (Point-to-Point) protokolom koji služi za modemsko povezivanje korisnika sa svojim ISP-om (Internet Service Provider). Usprkos čvrstoj povezanosti sa PPP-om, ne postoje objektivni razlozi zbog kojih CHAP ne bi mogao raditi i iznad drugih protokola.
Nesigurnosti CHAP protokola ne ovise o sloju iznad kojeg se CHAP nalazi. CHAP je ranjiv bez obzira da li se koristi sa PPP-om za dial-up vezu prema ISP-u, ili se koristi sa PPTP-om (Point-to-Point Tunneling Protocol) da zaštiti bežične mreže.
Ovaj dokument obraditi će PPP+CHAP preko IP-a, pošto su tehnike lažiranja IP adresa, kao i tehnike snifanja mrežnog prometa vrlo dobro poznate.

2. Način rada

Način CHAP autentikacije opisan je u RFC 1994. Proces autentikacije sa poslužiteljem koji autenticira klijenta ukratko je opisan u sljedećim koracima:

- nakon što je prošla faza uspostave veze, poslužitelj šalje *challenge* poruku klijentu,
- klijent u odgovoru šalje vrijednost izračunatu korištenjem jednosmjerne *hash* funkcije,
- poslužitelj provjerava odgovor u odnosu na vlastito izračunatu vrijednost *hash* funkcije; ukoliko se dobivene vrijednosti podudaraju, autentikacija je prihvaćena, dok bi se u suprotnom slučaju veza MORALA prekinuti,
- u slučajnim vremenskim intervalima, poslužitelj šalje nove *challenge* poruke klijentu, a prethodni niz koraka se ponavlja.

Prva tri koraka predstavljaju *challenge/response* proceduru, te osiguravaju zadovoljavajuću sigurnost ukoliko su ispunjeni sljedeći uvjeti:

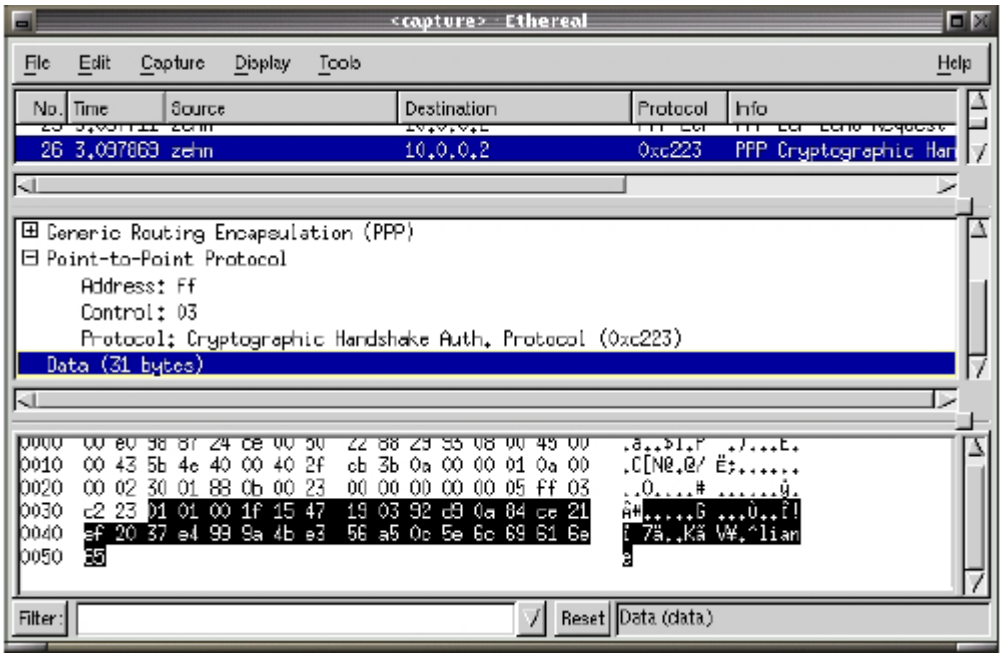
1. Korištena *hash* funkcija je kriptografski jaka. Uobičajene funkcije jesu MD5 ili SHA1.
2. Tajna vrijednost koja se koristi za izračun odgovora je jaka (ne pojavljuje se u rječnicima i sl.).
3. *Challenge* poruka koja se šalje klijentu je zaista slučajna, odnosno nije predvidljiva napadačima. Također se pretpostavlja da ista *challenge* poruka neće biti poslana unutar zadanog vremenskog perioda (npr. nakon restarta računala), što implicira da je skup slučajnih vrijednosti dovoljno velik.

Četvrti RFC zahtjev odnosi se na zaštitu kanala od preuzimanja (ukoliko napadač preuzme komunikacijski kanal, on neće znati odgovor na sljedeću *challenge* poruku i biti će odspojen). Ovaj zahtjev trebao je predstavljati specijalnu zaštitu CHAP protokola, no pokazuje se da on čini upravo suprotno.
Challenge/response protokoli sami po sebi su prilično slabe autentikacijske metode. Način njihovog rada je sličan kao da se napadaču da pristup `/etc/shadow`, što treba izbjegavati.

2.1. CHAP paketi iznutra

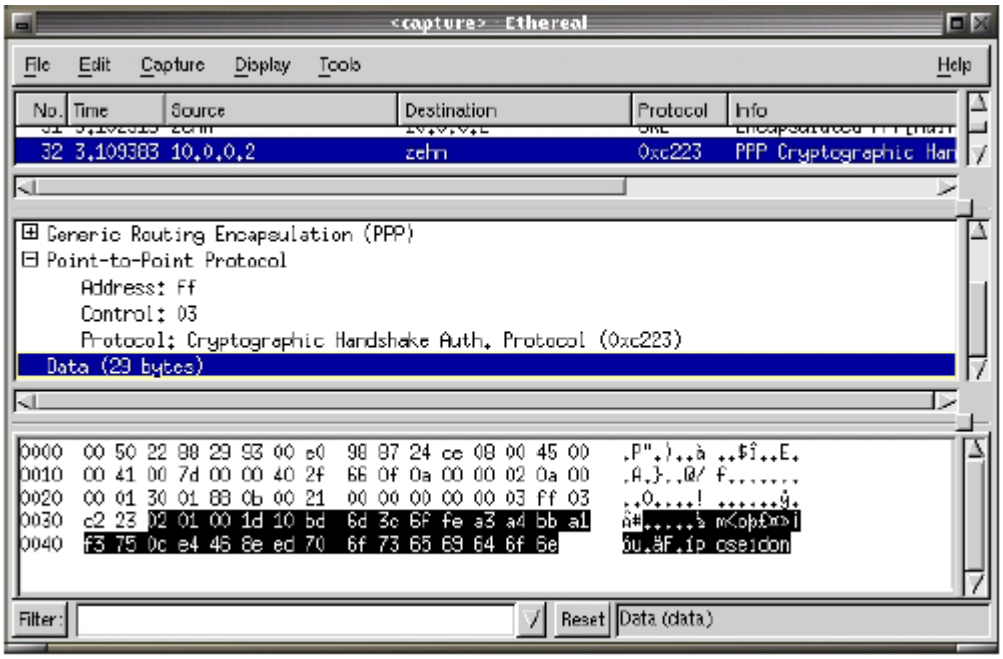
Slika 1 prikazuje uhvaćenu *challenge* poruku. Označeni podaci, prema RFC-u predstavljaju:

- 1 oktet kod; 01 za *challenge*, 02 za *response*,
- 1 oktet identifikator,
- 2 okteta, duljina CHAP odsječka,
- 1 okteta, duljina samog *challenge* ili *response* dijela,
- challenge/response dio,
- u slučaju *challenge* (kod 01) poruke ime poslužitelja, u slučaju *response* (kod 02) poruke korisničko ime.



Slika 1: *Uhvaćeni CHAP challenge paket*

U sjednici koja služi kao primjer vidi se *challenge* paket sa računala "*zehn*" prema adresi 10.0.0.2. Poslužiteljsko ime je "liane" (zadnjih 5 okteta paketa), dok je sama *challenge* fraza dugačka 16 okteta. Prvi paket od 10.0.0.2 prema *zehn*-u inicira komunikaciju i zahtijeva MD5 *hash* funkciju za obradu *challenge* fraze. Obje strane računaju *hash* vrijednost na isti način: povezuju identifikator (2. oktet paketa) sa tajnom frazom i challenge frazom te računaju vrijednost MD5 funkcije. Tajna fraza se određuje iz odgovarajuće datoteke koja sadrži tajne fraze (zaporke).
Kada klijent izračuna odgovor (*response*) šalje ga poslužitelju. Paket sadrži 16 okteta odgovora i korisničko ime koje poslužitelj treba koristiti za pronalaženje tajne fraze. Ukoliko klijent pošalje ispravan odgovor poslužitelj šalje paket o uspješnoj uspostavi veze. Slika 2 prikazuje *response* paket.



Slika 2: *Uhvaćeni CHAP response paket*

3. Otvaranje vrata

Unatoč činjenici da je CHAP dizajniran za slanje što manje količine informacija preko veze, mogu se dohvatiti sljedeći podaci:

- korisničko ime,
- ime poslužitelja,
- IP adrese klijenta i poslužitelja,
- ID korišten za izračun *response* fraze,
- *challenge* fraza i odgovarajuća *response* fraza.

Najvažnija stvar koja nedostaje jest tajna fraza. Ovaj način komunikacije zaista tajnu frazu drži tajnom.

Najjednostavniji način napada jest pokuša napad korištenjem rječnika, pošto sve nužne informacije za izračun *response* fraze postoje, odnosno jedina informacija koja je nepoznata jest tajna fraza. Pretpostavka je da je tajna fraza dovoljno dobra (što vjerojatno i nije slučaj, pa se isplati ovakav pokušaj napada). Nadalje, *challenge* fraza je vjerojatno jedinstvena i generator slučajnih vrijednosti dovoljno dobar, tako da ostali napadi sa ponavljanjem poruka ili slični neće biti od koristi.

Ponovni pogled na protokol izolira četvrti korak RFC specifikacije: klijent MORA odgovoriti na bilo koju *challenge* frazu koju primi. Što ukoliko se zahtjev pošalje poslužitelju, pričeka *challenge* fraza i zatim pusti već autenticiranog klijenta čija je komunikacija upravo uhvaćena da izračuna odgovor?

Analiza koda *pppd*-a (program koji je na obje strane odgovoran za CHAP autentikaciju) pokazuje da gornji slučaj funkcionira. Poslužitelj ne šalje ponovljene *challenge* fraze, ali klijent će odgovoriti bilo na bilo koji *challenge* koji primi.

Pošto PPP protokol ne zna ništa o IP adresama, ne postoji način da *pppd* odredi tko je tražio odgovor, pa bez obzira na zahtjev odgovara. Teoretski dakle, *pppd* odgovara na svaki zahtjev. U praksi su stvari složenije, pošto su PPP paketi umotani u PPTP okvire, jer se PPTP koristi za tuneliranje PPP-a preko IP-a. PPTP koristi sekvencijalne brojeve i adrese izvora/odredišta da bi odgovarajući *daemoni* mogli prihvatiti dolazne pakete. To znači da klijent sa adresom 10.0.0.2 neće odgovoriti na zahtjeve sa adrese 10.0.73.50, nego samo sa adrese 10.0.0.1 koji je legitimni PPTP poslužitelj u ovoj konfiguraciji. Sada je razumljivo zašto se koristi IP. Trivijalno je klijentu poslati lažiranu *challenge* poruku, koja će izgledati kao da je poslana sa originalnog poslužitelja. Klijent izračunava odgovor i ponovno ga šalje poslužitelju. To inducira dvije stvari, prvo moguće je uhvatiti *response* poruku i koristiti je za vlastitu autentikaciju, a kao drugo, poslužitelj također dobiva paket od legitimnog klijenta na koji on odgovara sa paketom o uspjehu kako se traži u RFC-u.

PPTP tunel također koristi 32-bitne sekvencijalne brojeve za bilježenje već poslanih paketa. Ovo nije veliki problem pošto je moguće uhvatiti posljednji važeći sekvencijalni broj. Stvari su još jednostavnije pošto *pptp* program prihvaća bilo koji sekvencijalni broj koji je veći od posljednjeg (što znači da je sigurno poslati PPTP paket sa sekvencijalnim brojem `0x00ffffff`). Korištenjem tog sekvencijalnog broja može se biti osigurati da će biti veći od posljednjeg valjanog sekvencijalnog broja koji je poslužitelj primio od klijenta.

3.1. Precizno podešavanje

Slika 3 pokazuje uspješno iskorištavanje nedostataka CHAP protokola na LAN mreži. Prvi korak je sakupljanje valjanih klijentskih i poslužiteljskih IP adresa i para korisničko ime/ime poslužitelja. Posebni *pppd* koristi te informacije za prijavljivanje na VPN bez poznavanja tajne fraze koja je povezana sa korisničkim imenom koje koristi. Dodatno, *challenge* i *response* poruke se pohranjuju što omogućava kasniji napad korištenjem rječnika.

Posljedica posebno prilagođenog *pptpd* daemona jest da klijenti žrtve gube vezu sa poslužiteljem, pošto *pptpd* daemon koji se nalazi na poslužitelju odbacuje sve pakete legitimnog klijenta jer je njegov sekvencijalni broj premalen da bi bio prihvaćen. Ovo se može izbjeći praćenjem poslanih sekvencijalnih brojeva, te odabiranjem posljednjeg uvećanog za jedan. Na taj način, klijent će propustiti samo jedan paket od poslužitelja. Drugi način je ponovno odbrojavanje sekvencijalnih brojeva (npr. slanjem `MAXSEQ-1` u lažnom zahtjevu i nakon dobivanja odgovora, slanjem `MAXSEQ` paketa tako da se sekvencijalni brojevi počnu inkrementirati od početka). Svi sljedeći paketi sa klijenta imaju veće sekvencijalne brojeve i valjani su.

Slika 3: *Uspješna prijava bez poznavanja tajne fraze*

4. Zaključak

Ovaj dokument pokazuje da CHAP protokol nije prava metoda autentikacije na IP mrežama. Iako nije pokazano kako se može postići zaobilaženje autentikacije na *dial-up* vezama, pošto nedostaju mogućnosti za snifanje/lažiranje, to ne znači da su te veze sigurne.

CHAP autentikaciju nikako ne bi trebalo koristiti u IP mrežama, a posebno ne na bežičnim LAN-ovima. U tim slučajevima valja razmotriti jače autentikacijske sheme kao što su RSA autentikacija ili Kerberos.