SIGURNOSNI PROPUSTI WEP PROTOKOLA
WEP PROTOCOL SECURITY VUNERABILITIES

Dejan Stjepanović, *Administrativna služba Grada Banja Luka*
Goran Prlina, *Administrativna služba Grada Banja Luka*

Sadržaj *– Bežični LAN je posljednjih godina postao široko rasprostranjen kako u poslovnim sredinama tako i u domaćinstvima, školstvu itd. U odnosu na žičane mreže, sigurnost bežičnih mreža je znatno komplikovanija. Jedan od sigurnosnih standarda u okviru WLAN standarda je Wireled Equivalent Privacy (WEP) koji se u posljednjih nekoliko godina pokazao kao nedovoljno siguran. U ovom radu su kritički sagledane sigurnosni propusti WEP-a, uključujući slabosti kratkih inicijalizacionih vektora i njihovog ponavljanja i slabosti RC4 kriptografskog algoritma. U radu će se predstaviti načini na koji se može testirati nivo sigurnosti WLAN-ova koji koriste WEP.*

Abstract – *The Wireless LANs have recently become widely deployed in business environments, at homes, at schools etc. Unlike the relative simplicity of wired Ethernet, WLAN security is much more complicated. Wireled Equivalent Privacy (WEP) is one of security standards within WLAN standard, that has been proven in recent years to be insufficiently secured. This paper critically reviews security flaws of WEP, including short IVs, key reuse, and inappropriate RC4 algorithms. In this paper we will show how to test WLAN security that use WEP.*

1. UVOD

Wired Equivalent Privacy (WEP) je sigurnosni prokol, specificiran u okviru IEEE Wireless Fidelity (Wi-Fi) standarda 802.11b. Kada se pojavio 1997. godine WEP je trebao da bežičnim mrežama (WLAN) obezbijedi nivo sigurnosti i privatnosti koju pruža standardni žičani LAN. Počevši od 2001. godine kriptoanalitičari su ukazali na ozbiljne slabosti WEP-a, koje su rezultovale tome da danas postoje metode i softver koji omogućavaju da se WEP konenekcija razbije za nekoliko minuta. 2003. godine IEEE komitet je predstavio novi, sigurniji protokol WPA. Godinu dana kasnije WEP protokol je proglašen prevaziđenim.

I pored svojih dokazanih slabosti WEP protokol se i danas masovno koristi. U gradu Banjaluci smo na nekoliko lokacija istražili sve dostupne bežične mreže. Od približno sto detektovanih mreža, gotovo jedna trećina je za zaštitu koristila WEP.

Naša namjera je bila da proučimo kako u WEP-u funkcioniše šifrovanje i dešifrovanje, koje su slabosti ovog protokola i kako funkcioniše njegovo razbijanje. Testiranje razbijanja WEP-a smo radili u laboratorijskim uslovima u WLAN-u koji smo sami postavili.

2. WIRED EQUIVALENT PRIVACY (WEP) PROTOKOL

WEP protokol je zasnovan na dijeljenom tajnom ključu koji se nalazi i na pristupnoj tački (access point – AP) i na svim klijentima koji pristupaju WLAN-u. Standardni 64-bitni WEP koristi 40-bitni tajni ključ. Ova relativno kratka dužina ključa je posljedica ograničavanja kriptografskih tehnika koju je u vrijeme definisanja WEP standarda sprovodila Vlada SAD. Kratki ključevi su podložni napadima brutalne sile. Nakon podizanja zabrane, većina proizvođača mrežne opreme je implementirala prošireni 128-bitni WEP protokol koji koristi 104-bitni ključ, koji je smanjio praktičnost upotrebe napada brutalne sile. [5]

WEP za tajnost podataka koristi protočni šifrator RC4, o kome će više riječi biti u narednom poglavlju. Za svaki paket koji se šifruje WEP generiše pseudo-slučajni 24-bitni inicijalizacioni vektor (IV). Standardni 64-bitni WEP koristi 40-bitni ključ koji se spaja sa 24-bitnim IV-om i tako se dobija 64-bitni ključ RC4 za šifrovanje. Kod 128-bitnog WEP-a, 24-bitni IV se dodaje na 104-bitni ključ da bi se dobio 128-bitni RC4 ključ za šifrovanje. Osnovna svrha IV-a je korištenje različitih ključeva za šifrovanje svakog paketa koji se šalje kroz mrežu.

802.11b paket koji se šalje kroz mrežu se sastoji od zaglavlja i korisnog tereta. Zaglavlje sadrži informacije

specifične za 802.11b protokol: MAC adresa izvora/odredišta, tip okvira itd. Korisni teret sadrži IP zaglavlje, TCP zaglavlje i same podatke sadržaja.

Pošto se za svaki paket generiše novi IV na osnovu koga se dobija RC4 ključ za šifrovanje, potrebno je IV proslijediti prijemniku, da bi se paket mogao dešifrovati. IV se kroz mrežu šalje u obliku čistog teksta, odnosno u nešifrovanom obliku.

Slika 1: Grafički prikaz 802.11b paketa

Prvi korak u kriptovanju podataka je kreiranje ICV-a (Integrity Check Value - vrijednost za provjeru integriteta), koji predstavlja 32-bitni CRC korisnog tereta, odnosno poruke koja se šalje. ICV ne obezbjeđuje integritet poruke u kriptografskom smislu, već prestavlja samo jedan dodatni vid zaštite od slučajne promjene podataka u transportu. ICV se potom dodaje na kraju korisnog sadržaja.

RC4 algoritam na osnovu ključa (tajni ključ + IV) kreira nizovni ključ (*keystream*) koji se koristi za operaciju 'ekskluzivnog ili' (XOR) sa čistim tekstom i njenom CRC vrijednosti, čime se dobija šifrovani tekst.

Ovaj proces možemo predstaviti sljedećom formulom:

$$C = (M \cdot c(M)) \oplus RC4(IV \cdot k) = P \oplus RC4(IV \cdot k)$$

Gdje su vrijednosti:

C – kriptovana poruka
M – poruka, čisti tekst
c(M) – CRC vrijednost poruke M
P = M · c(M) – konkatenacija poruke i njene CRC vrijednosti
k – dijeljeni tajni ključ, koji se nalazi na AP i klijentu
IV – inicijalizacioni vektor
RC4(IV · k) – *keystream*
⊕ - XOR operacija

Proces dešifrovanja WEP paketa je u suštini suprotan proces od šifrovanja. Kriptovani tekst se XOR-uje sa *keystream*-om da bi se dobio originalni čisti tekst. [4]

$$P = C \oplus RC4(IV \cdot k)$$
$$= (P \oplus RC4(IV \cdot k)) \oplus RC4(IV \cdot k) = P$$

3. RC4

WEP za kriprovanje koristi veoma rasprostranjeni RC4 protočni šifrator (*stream cipher*), koji se između ostalog koristi i za SSL. RC4 generiše pseudoslučajni niz bitova, tj. *keystream*, koji se XOR-uje sa čistim tekstom da bi se dobila šifrovana poruka. Za generisanje *keystream*-a šifrator koristi 2 algoritma:

1. Key Scheduling Algorithm (KSA)
2. Pseudo Random Generation Algorithm (PSGA)

Key Scheduling Algorithm (KSA) je prva faza u procesu enkripcije. Algoritam KSA izgleda ovako:

```
for i from 0 to 255
   S[i] := i
endfor
j := 0
for i from 0 to 255
   j := (j + S[i] + key[i mod keylength]) mod 256
   swap(&S[i],&S[j])
endfor
```

U prvoj petlji se generiše niz *S* koji sadrži vrijednosti od 0 do 255. U drugoj petlji segeneriše pseudoslučajni broj *j*, koji se dobija spajanjem vrijednosti niza *S* i tajnog ključa, dok se ujedno zamjenjuju bajtovi na pozicijama *i* i *j* u nizu *S*. Tajni ključ u slučaju WEP-a se sastoji od inicijalizacionog vektora i dijeljenog ključa.

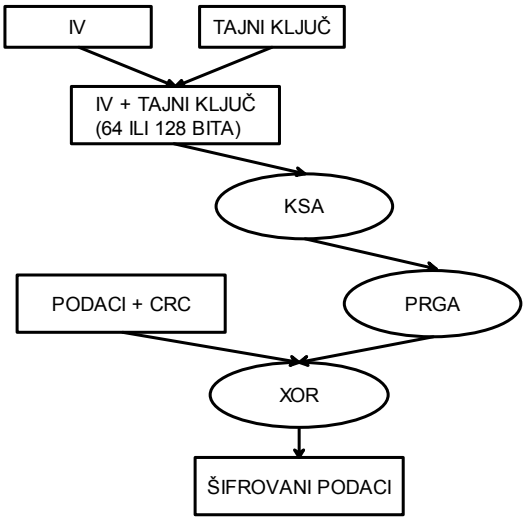
U sljedećoj fazi se pomoću KSA pseudoslučajnog niza *S* kreira *keystream* za šifrovanje podataka koji se šalju. Za tovaj korak se koristi Pseudo Random Generation Algorithm (PRGA), čiji algoritam izgleda ovako

```
i := 0
j := 0
while GeneratingOutput:
   i := (i + 1) mod 256
   j := (j + S[i]) mod 256
   swap(&S[i],&S[j])
   byte_cipher := S[(S[i] + S[j]) mod 256]
   result_ciphered := byte_cipher XOR byte_message
endwhile
```

Proces traje dok god ima podataka, a u slučaju WEP-a to je do kraja paketa sa podacima, odnosno oko 1500 bajtova. Za svaku iteraciju, PRGA inkrementira *i*, dodaje vrijednost niza *S* sa indeksom *i* na vijednost *j*, zamijenjuje vrijednosti *S[i]* i *S[j]*, i na izlazu daje vijednost *S* na lokaciji *S[i] + S[j] (mod 256)*. Svaka vrijednost *S* se zamijenjuje bar jednom svakih 256 iteracija. [6]

Sada ćemo rezimirati proces šifrovanja i dešifrovanja podataka u okviru WEP protokola. Za kriptovanja na prijemniku prvo se generiše IV, koji se potom spaja sa tajnim ključem. Ključ se koristi u KSA da se generiše pseudo-slučajni niz stanja, koji se onda koristi u PRGA da se kreira *keystream* koji se XOR-uje sa podacima u obliku čistog teksta i njene CRC vrijednosti. Takvi kriptovani podaci se šalju prijemniku gdje se po prijemu dekriptuju.

Kada podaci stignu do pijemnika, IV, koji je poslan kao čisti tekst pridružen kriptovanom tekstu, se izdvaja i spaja sa sa dijeljenom šifrom, da bi se dobio isti tajni ključ koji se koristio u procesu kriptovanja. Na isti način kao i u procesu šifrovanja se dobija *keystream* koji se XOR-uje sa kriptovanim tekstom. Time se dobija čisti tekst i njegova CRC vrijednost. CRC se uklanja i izračunava se novi CRC, koji se poredi sa originalnim CRC-om. Podaci se potom prihvataju ili odbacuju, u zavisnosti od toga da li se dvije CRC vrijednosti podudaraju ili ne.



Slika 2: Upošćena šema procesa šifrovanja

4. SLABOSTI WEP-A

Pošto je WEP baziran na RC4 enkripciji, naslijedio se i sve njegove nedostatke. IV se šalje kao obični tekst sa kriptovanim podacima, što znači da je dostupan svakome ko osluškuje radio talase. Znači, svakome su dostupni prvih 24 bita tajnog ključa. Svrha IV je izbjegavanje ponavljanja ključeva šifrovanja, međutim 24 bita nisu dovoljna da bi se to postiglo. Postoji vjerovatnoća od 50% da će se IV ponoviti poslije 5000 paketa. Nešto kasnije u ovom poglavlju ćemo detaljnje opisati opasnosti ponavljanja IV-a.

Dokazano je da su pojedini IV podložni probijanju, i oni se nazivaju slabi IV. Njihov format je (n + 3, 255, x) gdje je *n* bajt tajnog ključa koji se razbija. Vrijednot 255 indicira da je KSA u ranjivoj poziciji u algoritmu, a vrijednost *x* može biti bilo koja vrijednost.

RC4 algoritmi KSA i PRGA imju nekoliko ozbiljnih nedostataka. Oni puštaju informacije tokom prvih nekoliko iteracija u svojim algoritmima. Vrijednost *i* će uvijek biti 1, a vrijednost *j* će uvijek biti jednaka *S[1]* u prvoj iteraciji PRGA, a KSA se lako duplicira za prve 3 iteracije zbog činjenice da su poznata prva 3 karaktera, odnosno 24 bita ključa. Zatim, postoji vjerovatnoća od 5% da se vrijednosti u *S[0]* do *S[3]* neće promijeniti u prve 3 iteracije u KSA.

Nedoastatak WEP-a je i upotreba operacije 'ekskluzivno ili' (XOR) koja je relativno jednostavna, i koja omogućava da se izračuna bilo koja od 3 vrijednosti, ako su 2 vrijednosti već poznate.

Pokazalo se da je od presudnog značaja za razbijanje WEP-a, poznavanje određenih vrijednosti u zaglavlju poruke, koji se nikad ne mijenjaju. Prva vrijednost kriptovanih podataka je uvijek SNAP zaglavlje koje je jednako „AA", što znači da se XOR-ovanjem prvog bajta kriptovanih podataka može dobiti prvi izlazni bajt iz PRGA.

RC4 je protočni šifrator, a dobro poznti nedostatak protočnih šifratora je činjenica da korištenje istog *keystream*-a za kriptovanje dvije poruke može otkriti informacije o obje poruke. Ako sa C1 i C2 označimo dvije kriptovane poruke, a sa P1 i P2 njihove nekriptovane vrijednosti:

$C1 = P1 \oplus RC4(IV \cdot k)$
$C2 = P2 \oplus RC4(IV \cdot k)$

Možemo lako dokazati da XOR-ovanje dva šifrovana teksta (C1 i C2) poništava *keystream*, a rezultat je XOR dva čista teksta (P1 ⊕ P2). [4]

$C1 \oplus C2 = (P1 \oplus RC4(IV \cdot k)) \oplus (P2 \oplus RC4(IV \cdot k)) = P1 \oplus P2$

Drugim riječima, ponvljanje *keystream*-a omogućava više vrsta napada. Uspecijalnom slučaju, ako znamo jedan čisti tekst, drugi tekst se lako može otkriti. Generalno gledajući, u realnom slučaju, čisti tekstovi mogu imati dovoljnu redundanciju da se može otkriti i P1 i P2 na osnovu P1 ⊕ P2. Za ovo postoji nekoliko tehnika. Jedna tehnika je otkrivanje tekstova koji XOR-ovanjem daju P1 ⊕ P2. Ukoliko imamo *n* poruka koje su kriptovane istim *keystream*-om, imamo problem dubine *n*. Čitanje teksta u dubinu postaje lakše kako se *n* povećava, pošto se XOR svakog para čistog teksta može izračunati. Za rješavanje ovih problema postoje mnoge klasične tehnike kao što je anliza frekvencije i sl.

Da bi se izbjegli ovi napadi, WEP za svaki paket koji se kriptuje koristi drugi IV. I kao što smo već objasnili, taj IV se šalje u nekriptovanom obliku u sastavu podataka koji se šalju, što znači da je IV poznat i napadačima. Međutim WEP ne obezbjeđuje sigurnost od napada koji su bazirani na ponovnoj upotreni *keystream*-a.

Jedan od razloga za ponavljanje upotrebe *keystream*-a je nepravilno upravljanje IV-ovima. Pošto se tajni ključ u većini slučajeva rijetko mijenja, samim tim i ponovna upotreba IV znači i ponovna upotreba *keystream*-a. Dakle, ponovna upotreba IV izlaže sistem napadima ponovne upotrebe *keystream*-a (Key Reuse Attack - KRA).

WEP standard predlaže (ali ne zahtjeva) da se IV mijenja sa svakim paketom, ali ništa ne govori o načinu izbora IV-ova. U praksi se pokazalo da je kod mnogih bežičnih kartica upravljanje IV jako loše. U većini slučajeva IV-ovi se

resetuju na 0 svaki put kad se reinicijalizuju, da bi se zatim inkrementirali za svaki paket koji se šalje. Ovo znači da se *keystream*-ovi koji odgovaraju IV sa niskim vrijednostima često ponavljaju.

Ozbiljniji propust WEP standarda su veoma kratki IV-ovi u svim implementacija WEP standarda. Kao što je već rečeno, dužina IV je 24 bita, što znači da će se IV sigurno ponavljati. Ako pretpostavimo da prosječan access point šalje pakete dzužine 1500 bajtova pri prosječnoj brzini od 5Mbps, dolazi se do zaključka da će potrošiti sve IV za manje od pola dana. Ovaj propust je fundamentalan za sve WEP implementracije i ne može se izbjeći. [4]

U zavisnosti od impelemtacije ponovna upotreba ključa se može desiti znatno češće. Za implementaciju koja koristi slučajni IV za svaki paket, može se očekivati da se IV ponovi poslije samo 5000 paketa, što u relnom slučaju znači nekoliko minuta transmisije. Da situacija bude još gora, 802.11b standard ne zahtjeva da se IV mijenja sa svakim paketom, što znači da je moguće da se IV koristi za svaki paket koji se šalje, bez rizika nekompatibilnosti sa standardom.

Kada se otkriju dva kriptovana paketa koja koriste isti IV, mogu se iskoristiti mnogi napadi za otkrivanje čistog teksta, a ako je sadržaj jednog čistog teksta poznat, drugi se lako otkriva.

Postoje mnogi načini za otkrivanje potencijalnih kandidata za čisti tekst. Mnoga polja IP saobraćaja su predvidljiva, pošto protokoli koriste dobro poznate strukture u porukama, a sadržaj tih poruka je često predvidljiv. Na primjer, sekvenca za logovanje je često uniformna, pa je sadržaj upit za šifru poruke za logovanje isti za sve korisnike, i time poznat i samom napadaču. Kao drugi primjer, moguće je prepoznati dijeljene biblioteke koje se prenose kroz mrežu analizom mrežnog saobraćaja.

Najjednostavniji način za otkrivanje čistog teksta je kod AP-ova kod kojih je onemogućena kontrola pristupa mreži. U ovom slučaju, napadač šalje *broadcast* podatke AP-u, podaci se prihvataju zato što nema kontrole pristupa. Potom se ti podaci retransmituju sa AP u kriptovanom obliku. Ovaj sistem je moguć u mrežama koji imaju klijente sa WEP podrškom za šifrovanje i one bez podrške, a *broadcast* paketi se moraju slati svim klijentima.

5. FLURER, MATIN I ŠAMIR (FMS) NAPAD

Skot Flurer, Icik martin i Adi Šamir su 2001. Godine obavili rad u kojem su opisali način probijanja WEP-a na osnovu statističke analize. FMS napad se oslanja na upotrebu slabih IV-ova koje koristi RC4. Kod slabih IV-ova, napadač može na osnovu *m*-tog bajta *keystream*-a otkriti *m+1* bajt zbog slabosti algoritma koji se korisit za generisanje *keystream*-a. Dakle, za uspiješan napad potrebno je pribaviti IV koji postavlja algoritam u stanje u kom je moguće izvući informacije o ključu. Flurer, Matin i Šamir slučajve koji sadrže takve IV nazivaju riješenim. Svaki riješeni paket otkriva informaciju samo za jedan bajt tajnog ključa.

Kao što je već rečeno, prvi karakter SNAP zaglavlja je uvijek poznat, a to je “AA”. Poznata nam je i njegova kriptovana vrijednost koja je dobijena prisluškivanjem, jednostavnim XOR-ovanjem te dvije vrijednosti dobijamo prvi bajt *keystream*-a.

Za početak je potrebno pribaviti IV sa vrijednošću (3, 255, x). Napadač koristi IV kao prva 3 elementa u nizu *K[],* potom puni niz *S[]* i zvodi prve 3 iteracije u KSA. U trećem koraku napadač može da otkrije četvrti bajt ključa koristeći keystream izlaz *O* računajući *(O - j - S[i]) mod n = K[i]*, sa vrijednošću i = 3. U ovom trenutku, napadač još nema četvrti bajt ključa. Algoritam ne generiše sljedeći bajt ključa, već generiše moguću vrijednost ključa. Skupljanjem mnogo WEP paketa, i ponavljajući ove korake, napadač može generisati veliki broj ovih vrijednosti. Ispravna vrijednost će se pojavljivati znatno češće od bilo koje druge. Napadač može odrediti vrijednost ključa prepoznavajući ovu vrijednost i birajući je kao sljedeći bajt. Napad se može nastaviti za određivanje petog bajta ključa. [1]

Napad za otkrivanje bajtova ključa se mora izvesti sekvencijalno, tj bajt po bajt. Pribavljanje odgovarajućih slabih IV gotovo sigurno neće biti u odgovarajućem redosljedu. Međutim, paketi se mogu pohraniti, i kad se skupi dovoljan broj paketa sa slabim IV, napad se može izvesti. Za razbijanje 128-bitnog WEP-a potrenno je od 4 do 6 milion paketa. Međutim u praksi nije uvijek mogu brzo i jednostavno prikupiti milione paketa. FMS napad je pasivni napad, koji samo osluškuje mrežni saobraćaj. Ukoliko na mreži nema dovoljno klijenata, ili ih uopšte nema, FMS napad je teško izvesti. [1]

6. PUŠKIN, TUS, VAJNMAN (PTW) NAPAD

Andrej Puškin, Erik Tus i Ralf-Filip Vajnman (PTW) su 2007. godine publikovali rad u kome su opisali novi način napada na WEP. Autori su iskoristili neke starije ranjivosti RC4. Napad funkcioniše tako što se prikupljaju specifični tipovi paketa, kod kojih se mogu otkriti više informacija o čistom tekstu.

Ako host A želi da pošalje IP datagram hostu B, potrebno je da zna fizičku adresu hosta B ili gateway-a kroz koji može doći do hosta B. Za razrješavanje IP adresa hostova u njihove fizičke adrese, koristi se *Address Resolution Protocol* (ARP). Ovo funkcioniše na sljedeći način: Host A šalje ARP zahtjev prema *link layer broadcast* adresi - ovo znači da host A traži fizičku adresu hosta B. Host B odgovara hostu A sa ARP odgovorom koji sadrži njegovu fizičku adresu. Pošto je ARP *link layer* protokol, on uglavnom nije ograničen bilo kojim filterima.

ARP zahtjevi i ARP odgovori su fiksne dužine . Pošto dužina paketa nije maskirana WEP-om, oni se lako mogu

razlikovati od drugog saobraćaja. Prvih 16 bajtova čistog teksta ARP-a se sastoje od 8 bajtova 802.11 LLC zaglavlja, nakon kojih slijedi 8 bajtova samog ARP paketa. LLC zaglavlje je fiksno za svaki ARP paket (AA AA 03 00 00 00 08 06). Prvih 8 bajtova ARP zahtjeva su također fiksni (00 01 08 00 06 04 00 01). Za ARP odgovor, posljednji bajt je 02, a ostatak bajtova je identičan ARP zahtjevu. ARP zahtjev se uvijek šalje *broadcast* adresi, dok se ARP odgovor šalje *unicast* adresi. Pošto fizičke adrese nisu kriptovane pomoću WEP-a, lako je razlikovati kriptovani ARP zahtjev od odgovora. XOR-ovanjem ARP paketa sa ovim fiksnim šablonima može se dobiti 16 bajtova *keystream*-a. [3]

Da bi se ubrzalo prikupljanje paketa, moguće je injektovati uhvaćene ARP zahtjeve natrag u mrežu. Destinacija odgovara sa ARP odgovorom. Ako su inicijator i destinacija originalnog ARP zahtjeva mrežni klijenti, svaki injektovani paket će generisati tri nova paketa, zato što će prenos paketa preusmjeriti AP. Pošto ARP odgovori ističu relativno brzo, obično je potrebno nekoliko sekundi da napadač uhvati ARP zahtjev i počne ga injektovati u mrežu. [2]

Moguće je čak i ubrzati vrijeme potrebno da se uhvati prvi ARP paket na taj način što se klijentu pošalje poruka o deautentifikaciji, koja mu govori da je izgubio kontakt sa AP. U nekim konfiguracijama klijenti se automatski spajaju na mrežu brišući svoj ARP keš. Prvi IP paket poslat sa klijenta će prouzrokovati ARP zahtjev za Ethernet adresom destinacije. [3]

Ovaj napad je pokazao da je dovoljno prikupiti 40.000 paketa da bi se sa vjerovatnoćom od 50% mogao otkriti tajni ključ, a sa 100.000 paketa ta vjerovatnoća je 100%. Ovo je ogroman napredak u odnosu na FMS napad.

7. SOFTVER ZA NAPAD

Za napad smo koristili softverski paket *Aircrack-ng,* kojeg čini 18 alata za detektovanje, prisluškivanje, analizu i razbijanje paketa bežičnih mreža. Paket radi sa mrežnim karticama čiji upravljački programi podržavaju *raw monitoring mode.* Paket je dostupan za *Linux* i *Windows* platformu. Većina *Windows* upravljačkih programa za mrežne kartice ne podržava *raw monitoring mode,* a njihov izvorni kod uglavnom nije javno dostupan, za razliku od Linux upravljačkih programa. Stoga je upotrebljivost ovog paketa veća na Linux platformi.

Mi smo koristili *BackTrack 4 live* Linux distribuciju, koja u sebi već sadrži *Aircrack-ng 1.0,* na MSI laptopu sa Intelovim *chipset*-om na inegrisanoj bežičnoj mrežnoj kartici. Za uspiješno razbijanje šifre bila su nam dovoljna samo četiri alata iz ovog paketa. *Airmon-ng* smo koristili za stavljanje bežične mrežne kartice u monitoring mod. *Airodump-ng* je *packet sniffer,* kojeg smo iskoristili za prisluškivanje mrežnog saobraćaja na sa AP-a. *Aireplay-ng* smo koristili za injekciju ARP paketa u mrežu, kako je to opisano u PTW napadu. Za uspiješno probijanje šifre bilo nam je potreno oko 17.000 paketa, koje smo prikupili za manje od 15 minuta. Za to smo koristili alat *aircrack-ng.* Probijanje šifre je trajalo nekoliko sekundi.

8. ZAKLJUČAK

Počevši od 2001. godine, različitim napadima je dokazano da WEP ne zadovoljava sigurnosne standarde u zaštiti podataka u bežičnim mrežama, a on je i dalje u širokoj upotrebi, iako je proglašen prevaziđenim protokolom. Naše društvo u posljednje vrijeme postaje sve više zavisno od bežičnih mreža, zato proučavanje njihovih nedostataka i sigurnosti dobija sve veći značaj.

Stoga WEP treba izbjegavati kad god je to moguće. Ukoliko mrežna oprema ne podržava WPA/WPA2, onda je WEP jedina opcija. Da bi se povećala sigurnost takvih mreža predlažemo da se AP podesi da bude sakriven, što znači da njegov SSID bude nevidljiv za okolinu. Iako ovo nije nikakva prepreka za alate za praćenje mrežnog sadržaja, jer sami paketi nisu skriveni, ali ovim se postiže da prosječan korisnik ne može detektovati WLAN.

Takođe predlažemo da se u mrežama gdje su poznati svi klijenti, podesi lista za kontrolu pristupa koja je bazirana na MAC adresama klijenata. Iako ni ovo ne predstavlja veliki problem za alate za razbijanje WEP-a, jer se MAC adrese mogu lako zamaskirati, ali ovim se dodatno povećava sigurnost. Kada je to moguće, mrežni saobraćaj bi trebalo preusmjeriti kroz servise kao što su SSH ili VPN, da bi se obezbjedio dodatni nivo kriptovanja.

LITERATURA

[1] Scott Fluhrer, Itsik Martin, Adi Shamir. *Weaknesses in the Key Scheduling Algorithm of RC4*

[2] Ross Buffington, Will Proffitt. *WEP (In)Security*

[3] Pyshkin, Tews, Weinmann. *Breaking 104 Bit WEP in less than 60 seconds*

[4] Nikita Borisov, Ian Goldberg, David Wagner. *Intercepting Mobile Communications: The Insecurity of 802.11*

[5] William A. Arbaugh, Narendar Shankar, Y.C. Justin Wan. *Your 802.11 Wireless Network has No Clothes*

[6] *http://www.informit.com/*