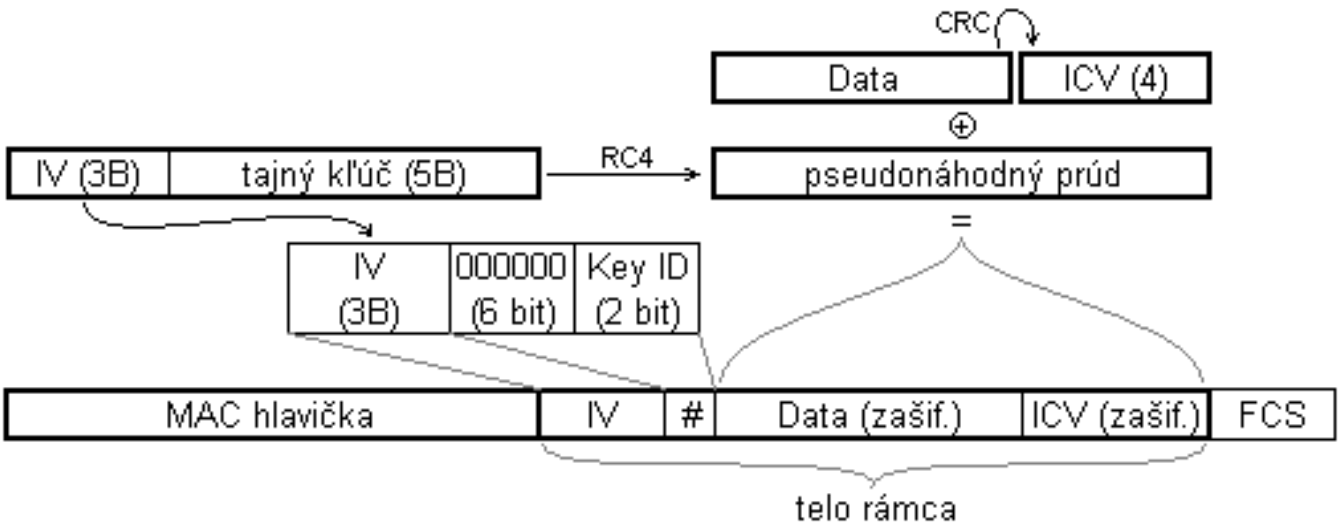
Bezpečnost a Hacking WiFi (802.11) - 3. WEP

Vložil/a matej [1], 22 Prosinec, 2009 - 17:34

- Cracking [2]
- Hacking [3]
- Networks & Protocols [4]
- Security [5]
- WiFi & Wireless [6]

WEP (Wired Equivalency Protection)

Protokol na zabezpečenie WLAN definovaný v prvom IEEE 802.11 štandarde [3] sa nazýva WEP – Wired equivalent privacy. Bol vymyslený na poskytnutie bezpečnosti úrovne drôtových sietí, ale kvôli slabému kryptografickému základu tomu tak nie je.



obr. 4 1: Enkapsulácia WEP

Použitie WEP je naznačené na obr. 4 1 a bolo už popísané v bakalárskej práci [7]. Na šifrovanie sa používa prúdová šifra RC4, na zabezpečenie obsahu ICV (Integrity Check Value, kontrolná hodnota integrity), vypočítané pomocou CRC-32 (Cyclic Redundancy Code, cyklický kód určený na detekciu chýb) a tiež zašifrované. Inicializačný vektor (IV) sa pripojí k tajnému kľúču (štyri definovateľné kľúče odlíšené pomocou Key ID) a použije na inicializovanie stavového poľa pre RC4 algoritmus. Proces inicializácie stavového poľa sa nazýva KSA (Key Scheduling Algorithm, algoritmus na rozvrhnutie kľúča). Za telom rámca nasleduje FCS (Frame Check Sequence, kontrolná hodnota rámca) vypočítané až hardvérovo. Útoky na WEP sú popísané v nasledujúcom texte a vyplývajú z týchto nedostatkov:

• použitie statického kľúča (maximálne 4 kľúče, statické), mení sa len IV;
• opakovanie IV (cyklus len 224);
• použitie rovnakého algoritmu na šifrovanie aj autentifikáciu;
• linearita CRC-32 a operácie XOR;
• šifrovanie ICV spoli s dátami;
• nedostatky použitého algoritmu RC4.

4.1 Brute-force

Známy začiatok plaintextu a krátka reálna dĺžka kľúča umožňuje pri nazbieraní malého množstva párov {IV, začiatok RC4 výstupu} urobiť výpočtovo náročný brute-force útok, respektíve slovníkový útok na zistenie zdieľaného šifrovacieho kľúča. Táto metóda je použiteľná len pre 64 bitové WEP – dĺžka tajného kľúča je iba 40 bitov, pri použití alfanumerických a tlačiteľných znakov je entropia kľúča oveľa menšia.

4.1.1 Útok na generátor kľúča

Mnoho ovládačov sieťových kariet umožňuje namiesto alfanumerického kľúča zadať tzv. „passphrase“, z ktorej sa generátorom vytvoria štyri kľúče. Tento generátor je bežne používaný, ale nie je nijak štandardizovaný. 64-bitová verzia využíva XORovanie (exclusive OR, vylučujúce alebo) jednotlivých znakov passphrase navzájom a RC4 PRNG (Pseudo-Random Number Generator, generátor pseudonáhodnej postupnosti čísel) takým spôsobom, že výsledný 40-bitový kľúč, bez ohľadu na dĺžku pôvodného passphrase, má entropiu iba 21 bitov. Detailne je tento generátor a útok naňho popísaný v [8]. Demonštračný program od Tima Newshama vie s pomocou 2 odchytených rámcov takýto WEP kľúč prelomiť na stroji P4 2.6 GHz do 10 sekúnd:

```
$ ./wep_crack -b wep64-passphrase-2packets-stripped.cap
success: seed 0x00327821,  [generated by AAAa`9sa]
wep key 1: da 37 11 e6 ac
wep key 2: 3b dd 3b c4 ef
wep key 3: 09 1d 2c c8 86
wep key 4: c6 09 e9 3e 90
834594 guesses in 3.57 seconds:  234024.09 guesses/second
```

4.1.2 Úplné prehľadávanie

Ak je pri 64-bitovom WEP použitý silný 40-bitový kľúč, ešte stále je možné ho hrubou silou zlomiť. Jon Ellch napísal na tento účel niekoľko programov:

• **jc wepcrack** – umožňuje distribuované lámanie
(približne 300 000 kľúčov/sek na jednom P4 3.6 GHz)
• **ps3 wepcrack** – lámanie na Sony PlayStation3 využívajúce 6 VPU (Vector Processing Unit, vektorové procesné jednotky) na doske
(približne 1 440 000 kľúčov/sek)
• **pico wepcrack** – hardvérovo akcelerované lámanie pomocou Pico karty, CardBus FPGA (Field-programmable gate array, programovateľné hradlové pole) od firmy Pico Computing (http://www.picocomputing.com/ [7])
(približne 9 000 000 kľúčov/sek)

Na jedinom notebooku s Pico kartou je teda možné úplné prehľadanie pre 40 bitový kľúč za necelých 34 hodín (najhorší prípad).

4.1.3 Obrana voči brute-force

Použitie 128 bitového WEP (dĺžka kľúča 104 bitov) s náhodne vygenerovaným kľúčom možnosť brute-force útoku značne minimalizuje. V praxi sa ale metóda brute-force samotná takmer nepoužíva, pretože existujú oveľa efektívnejšie spôsoby, ako WEP prelomiť (viď. ďalej v tejto kapitole). Hrubá sila sa obvykle používa len na dopočítanie 1 2 chýbajúcich bajtov kľúča pri FMS (Fluhrer-Mantin-Shamir, autori útoku) a KoreK útokoch (viď. 4.7 a 4.8).
Použitie RSN (WPA/WPA2) zabráni tomuto druhu brute-force útoku – šifrovací kľúč je dlhší a mení sa, a preto „nie je čo hľadať“.

4.2 Injekcia rámcov

Ochranu pred zdvojenými rámcami poskytuje obvykle firmware WLAN zariadenia, a to pomocou poľa Sequence number v hlavičke. WEP šifruje a zabezpečuje pomocou ICV iba dátovú časť rámcov. Špecifikácia WEP umožňuje opakovanie sa IV a kľúč je statický – je teda možné ľubovoľný zachytený rámec znovu vyslať. Aby nebol identifikovaný ako zdvojený, postačuje zmeniť Sequence number. Reinjekciou rámcov môžeme dosiahnuť rôzne ciele:

• marenie toku dát,
• celkové zvýšenie prevádzky na sieti za účelom zachytiť čo najviac rôznych IV pre FMS/KoreK útoky (viď. 4.7 a 4.8),
• zvýšenie ARP (Address Resolution Protocol, protokol na zisťovanie adries) prevádzky pre Kleinov útok (viď. 4.9).

4.2.1 ARP reinjekcia

ARP rámce sú ľahko identifikovateľné aj v zašifrovanej forme, pretože sú krátke, a ARP request (požiadavka) má cieľovú MAC adresu broadcast (FF:FF:FF:FF:FF:FF). AP takýto rámec prepošle ostatným STA; niektoré AP ho najprv dešifrujú a následne zašifrujú s novým IV. Pri zvyšovaní prevádzky sú dobré aj v tom, že cieľový adresát na ne promptne odpovie, čím vygeneruje nový rámec, s novým IV.
Program aireplay ng z balíka Aircrack-ng umožňuje v režime monitor reinjekciu rámcov podľa zadaných pravidiel. Pre zvýšenie ARP prevádzky ho spustíme ako root s parametrom -3 takto:

```
# ./aireplay-ng -3 -b 00:11:3b:07:00:14 -h 00:11:3b:0b:22:0c rausb0
The interface MAC (00:11:09:29:62:38) doesn't match the specified MAC (-h).
        ifconfig rausb0 hw ether 00:11:3B:0B:22:0C
Saving ARP requests in replay_arp-0502-004658.cap
You should also start airodump-ng to capture replies.
Read 51729 packets (got 49845 ARP requests), sent 51017 packets...(277 pps)
```

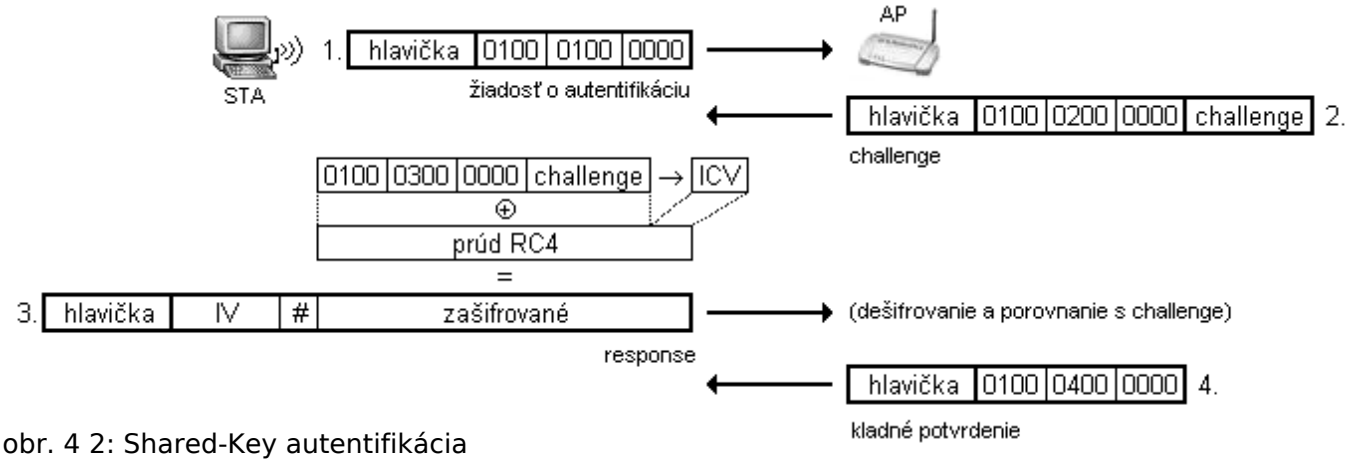
-3 určuje typ útoku: ARP reinjekcia,
-b ... určuje BSSID (MAC adresa AP),
-h ... určuje zdrojovú MAC adresu, ktorá sa má pre vyslané rámce použiť,
rausb0 je zariadenie použité na vyslanie rámca (musí byť v monitor mode).

Po zachytení rámca, o ktorom sa predpokladá, že je to ARP request, sa tento so zmenenou hlavičkou pošle, generujúc tak od adresáta skoro 300 ARP response (odpoveď) rámcov za sekundu, s novými IV. Ak žiadne ARP rámce nezachytíme, môžeme si ich skúsiť vynútiť pomocou deautentifikácie (bližšie k tomuto v 6.4).

4.2.2 Ochrana voči reinjekcii

Podľa IEEE štandardu nie je pre WEP zadefinovaná ochrana voči reinjekcii paketov. Niektoré zariadenia reinjekcii čiastočne zabraňujú tým, že IV prijatých rámcov si ukladajú do cache (vyrovnávacia pamäť) a ignorujú všetky rámce s rovnakým IV, akonáhle ich počet presiahne istú hranicu (napríklad 64).
Použitie RSN (Robust Security Network, sieť s robustnou bezpečnosťou) (WPA/WPA2) zabráni reinjekcii, pretože neumožňuje znovupoužitie IV a pomocou MIC je zabezpečená aj hlavička rámca. Alternatívou môže byť Wireless IDS (viď. 9.4).
Generovaniu ARP prevádzky na sieti je možné zabrániť statickými ARP tabuľkami na všetkých staniciach. To je však ťažko manažovateľné a zle škálovateľné riešenie.

4.3 Zbieranie slovníka PRGA pomocou Shared-Key autentifikácie



obr. 4 2: Shared-Key autentifikácia

Príklad úspešnej Shared-Key autentifikácie je na obr. 4 2. Štandard určuje, že challenge text (výzva) má mať dĺžku 128 znakov. Formát challenge elementu v druhom autentifikačnom rámci je {Element ID, Length, Challenge Text}, kde Element ID je 10h, Length 80h (128). Po odchytení druhého rámca a výpočte ICV teda poznáme celý plaintext (otvorený text), ktorý sa posiela zašifrovaný (ciphertext) v treťom rámci ako response (odpoveď), konkrétne 6+2+128+4 = 140 bajtov. Po odchytení tretieho rámca máme teda „úplne zadarmo“ pseudonáhodnú sekvenciu (ciphertext plaintext) dĺžky 140 pre dané IV (zvolí STA).
Autentifikácia prebieha len pri nadväzovaní konektivity, môžeme si ju ale vynútiť sfalšovanou deautentifikáciou (viď. 3.1.1 a tiež 6.4) a vytvoriť si tak slovník takej veľkosti, ako na konkrétny účel potrebujeme.

4.3.1 Využitie slovníka PRGA

Nazbieranú databázu dvojíc {IV, prúd PRGA} môžeme zneužiť viacerými spôsobmi, bez toho aby sme poznali WEP kľúč:

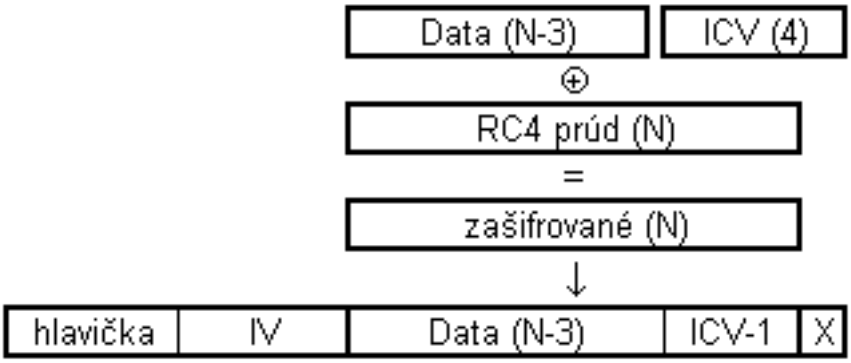
• autentifikácia do siete – môžeme sami použiť PRGA prúd (Pseudo-Random Generation Algorithm, algoritmus generovania pseudonáhodnej postupnosti) a byť tak autentifikovaným účastníkom;
• injekcia/posielanie rámcov – môžeme vyslať ľubovoľne zostrojený rámec, pretože ho sami zašifrujeme (viď. aj 4.2);
• dešifrovanie prijatých/zachytených rámcov – po vytvorení dostatočne veľkej databázy môžeme dešifrovať veľké množstvo rámcov, prípadne všetky rámce, ktorých dáta sú kratšie ako 140 bajtov (takýto útok je možný, ale zdĺhavý a v praxi sa nepoužíva);
• zväčšiť dĺžku známeho prúdu pomocou Arbaugh útoku (opísaný ďalej v 4.4) a vytvoriť si tak lepšiu databázu.

4.3.2 Obmedzenie zbierania slovníka PRGA

Ak chceme zabrániť zneužitiu Shared-Key autentifikácie, treba ju vypnúť na všetkých zariadeniach. Bohužiaľ niektoré ovládače nedokážu mať zapnuté šifrovanie WEP a pri tom Open System autentifikáciu (otvorený systém, prázdna autentifikácia).
Použitie RSN (WPA/WPA2) zabráni zneužitiu slovníka, šifrovací kľúč sa totiž mení a nie je možné znovupoužitie IV. Taktiež pri použití RSN sa robí Open System autentifikácia, a následne až po asociácii autentifikácia pomocou EAPOL (Extensible Authentication Protocol over LAN, rozšíriteľný autentifikačný protokol cez lokálnu sieť).

4.4 Indukčný útok Arbaugh

Tento útok publikoval William A. Arbaugh v máji 2001 v [9], umožňuje ľubovoľne predĺžiť známy RC4 prúd dĺžky N.



obr. 4 3: Indukčný útok Arbaugh

Z krátkych rámcov so známym predpokladaným plaintextom (napr. ARP, DHCP komunikácia) (Dynamic Host Configuration Protocol, protokol na dynamickú konfiguráciu účastníkov siete), alebo pomocou Shared-Key autentifikácie vieme získať N bajtov PRGA (RC4 prúdu) pre dané IV. Môžeme potom zostrojiť rámec s dátami dĺžky N-3, pre ktoré vypočítame ICV a pripojíme z neho iba 3 bajty – obr. 4 3. Pripojíme ďalší bajt s hodnotou X a rámec vyšleme.
V prípade, že tento rámec AP prepošle, resp. dostaneme na neho odpoveď (ak sme vyslali napríklad ICMP alebo ARP rámec), znamená to, že hodnota X je správna. N+1 vý bajt RC4 prúdu potom vypočítame ako X xor 4. bajt ICV. Pre X existuje najviac 256 možností, teda na získanie jedného bajtu potrebujeme vyslať najviac 256 rámcov. Získali sme teda správnych N+1 bajtov RC4 prúdu a indukčným spôsobom vieme pokračovať až do požadovanej dĺžky.

4.4.1 Zložitosť útoku Arbaugh

Náročnosť útoku je relatívne nízka, ak potrebujeme získať RC4 prúdy pre malé množstvo IV – pri 100 rámcoch za sekundu vieme získať prúd dĺžky 2400 bajtov (približne MTU (Maximum Transmission Unit, maximálna posielateľná veľkosť jednotky)) priemerne za 50 minút, v najhoršom prípade za 256*2400/100 sekúnd (t.j. 1.7 hod). Pre vytvorenie kompletného slovníka tento útok nie je vhodný. Praktická implementácia nie je známa, princíp bol však využitý pre chopchop útok (ďalej v 4.5).

4.4.2 Obmedzenie útoku Arbaugh

Ochrana voči Arbaugh útoku je rovnaká, ako voči reinjekcii (viď. 4.2.2) – potrebujeme zabrániť opakovaniu IV a falšovaniu rámcov – použitím RSN (WPA/WPA2) alebo neštandardne pomocou zahadzovania často sa opakujúcich rámcov s rovnakými IV na zariadení.

4.5 KoreK chopchop

Koncept chopchop („sek-sek“) útoku bol zverejnený v septembri 2004 na fóre netstumbler.org spolu s proof-of-concept utilitou. Nazýva sa aj „inverzný Arbaugh útok“. Umožňuje dešifrovať ľubovoľný zachytený rámec aktívnym iteratívnym spôsobom a získať tak jeho obsah (alebo aspoň väčšinu jeho obsahu) a zároveň použitý RC4 prúd pre dané IV.

4.5.1 Princíp chopchop útoku

Možnosť útoku spočíva v linearite RC4 šifrovania a CRC (ICV). Detailne je popísaný v balíku so zdrojovým kódom chopchop utility.
Odrežeme posledný bajt dátovej časti rámca, o ktorom predpokladáme, že bol napríklad 0, prepočítame ICV, a rámec (o 1 bajt kratší ako pôvodný rámec s neznámym obsahom) vyšleme. Ak ho AP prepošle, znamená to, že náš predpoklad bol správny a pokračujeme iteratívne. Ak nie, vyskúšame ďalšiu možnosť.

Niektoré druhy rámcov nie je možné dešifrovať celé, pretože po dosiahnutí malej dĺžky prestanú

dávať zmysel a nedostaneme žiadnu odozvu.
Aby celý proces bežal rýchlejšie, nebudeme čakať po každom rámci na odozvu, ale očíslujeme rámce pomocou cieľovej MAC adresy – podľa nej po spätnom prijatí rámca potom vieme, ktorý bajt a na ktorom mieste bol správny. Kratšie rámce tak môžeme dešifrovať za niekoľko sekúnd (viď. dalej).

4.5.2 Realizácia chopchop útoku

Pomocou monitorovacieho režimu zachytíme krátky zašifrovaný rámec. Pôvodná KoreK ova proof-of-concept chopchop utilita je hardvérovo závislá (použité ovládače linux-wlan-ng pre PrismII chipset) a preto nefungovala. Útok chopchop však podporuje aj balík Aircrack-ng, pomocou utility **aireplay-ng**. Príklad spustenia:

```
# ./aireplay-ng -4 -b 00:11:3b:07:00:14 -h 00:17:31:ba:ef:e4 -r x.cap rausb0
... (výpis rámca z x.cap, je to zašifrovaný ARP request)
Use this packet ? y

Saving chosen packet in replay_src-0505-171500.cap

Offset   85 ( 0% done) | xor = 4E | pt = 88 |  307 frames written in   923ms
Offset   84 ( 1% done) | xor = 95 | pt = A3 |  315 frames written in   943ms
Offset   83 ( 3% done) | xor = C4 | pt = F0 |  105 frames written in   315ms
... (postupný výpis všetkých bajtov)
Offset   34 (98% done) | xor = 2F | pt = 08 |  210 frames written in   629ms

Saving plaintext in replay_dec-0505-181002.cap
Saving keystream in replay_dec-0505-181002.xor

Completed in 32s (1.50 bytes/s)
```

-4 určuje typ útoku (KoreK chopchop),
-b ... určuje BSSID (MAC adresa AP),
-h ... určuje zdrojovú MAC adresu, ktorá sa má pre vyslané rámce použiť,
-r ... určuje pcap súbor, z ktorého sa má načítať zašifrovaný rámec,
rausb0 je zariadenie použité na vyslanie rámca (musí byť v monitor mode).

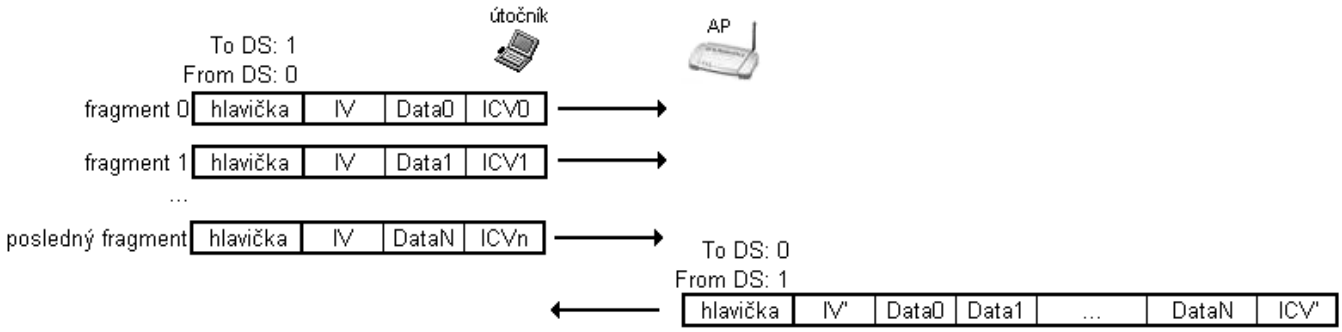
Dešifrovaný rámec aireplay-ng uloží do nového pcap súboru a navyše získame jednu RC4 sekvenciu (.xor súbor), ktorú môžeme použiť napríklad na injekciu rámcov (viď. 4.2) apod. Navyše je útok bežným používateľom nespozorovaný, pretože rámce s nesprávnym ICV sa zahodia.

4.5.3 Obmedzenie chopchop útoku

Ako už bolo spomínané vyššie v 4.2.2 Ochrana voči reinjekcii, 4.4 Indukčný útok Arbaugh, snahou je zamedziť opakovaný výskyt veľkého množstva rámcov s rovnakým IV – je to síce neštandardizovaný spôsob, ale funguje efektívne. Niektoré AP nie sú voči tomuto útoku náchylné vďaka zahadzovaniu rámcov kratších ako 60 bajtov, ostáva však možnosť dešifrovať koniec väčších rámcov.
Použitie RSN úplne zabráni tomuto útoku, pretože nie je možné opakovať IV. Alternatívou môže byť Wireless IDS (viď. 9.4).

4.6 Fragmentačný útok

V septembri 2005 predstavil Andrea Bittau v [10] praktický fragmentačný útok.



obr. 4 4: Fragmentačný útok

Jeho princíp spočíva práve v defragmentácii. Ak vyšleme K fragmentovaných rámcov (K=N+1 podľa obr. 4 4) do distribučného systému, AP tieto fragmenty pospája a pošle v jednom rámci (až do veľkosti svojej MTU, resp. MTU použitého protokolu vyššej vrstvy).
Keď je na sieti použitý WEP, jednotlivé rámce zašifrujeme pomocou známej dvojice {IV, RC4 prúd} do distribučného systému (To DS bit=1). AP ich defragmentuje, zašifruje pomocou IV', a ak cieľová MAC adresa nie je určená pre inú sieť, pošle nazad (From DS bit=1, do vzduchu pre známeho alebo neznámeho adresáta), ako je naznačené na obr. 4 4. Plaintext zašifrovaného defragmentovaného rámca sme však zvolili my (Data0 až DataN), a teda vieme hneď určiť novozískanú dvojicu {IV', dlhší iný RC4 prúd}.
Jednou z utilít, ktoré fragmentačný útok implementujú, je **aireplay-ng** z balíka Aircrack-ng, a to s parametrom -5. Vytvára rámce dlhé 35 bajtov (s 3 bajtami dát na rámec). Použitý AP však odmietol takéto krátke fragmenty skombinovať, a preto útok nezafungoval – bolo by nutné utilitu upraviť pre použitie dlhších fragmentov, čo ale znamená potrebu dlhšieho známeho plaintextu a celý koncept tým stráca zmysel.

4.6.1 Výhody fragmentačného útoku

Tento útok je oveľa efektívnejší ako Arbaugh alebo chopchop (opísané v 4.4 a 4.5), pretože nepotrebuje posielať skusmo neplatné rámce. Teoreticky umožňuje už s 5 známymi bajtami PRGA (1 bajt pre dáta a 4 pre ICV) poslať ľubovoľne dlhý rámec. V bežnej prevádzke vieme pomerne spoľahlivo odhadnúť až 7-16 bajtov plaintextu, a teda aj PRGA – podľa veľkosti rámca určíme protokol vyššej vrstvy a podľa MAC adries z hlavičky rámca môžeme odhadnúť niektoré z polí hlavičky protokolu vyššej vrstvy (ARP, ICMP, IP, ...).
Získané pseudonáhodné sekvencie môžeme potom použiť na injekciu rámcov (vyššie v 4.2), prípadne zostavenie kompletného PRGA slovníka (odhadom 1 deň floodovania).

4.6.2 Obrana voči fragmentačnému útoku

Ako už bolo spomenuté vyššie, AP a stanice, ktoré neprijímajú krátke fragmenty, nie sú náchylné voči tomuto útoku. Ďalšou obranou je obmedzenie opakovania sa IV (viď. aj 4.2.2 Ochrana voči reinjekcii) buď zahadzovaním často sa opakujúcich rámcov s rovnakým IV, alebo použitím RSN (viď. kapitola 5. WPA a WPA2).

4.7 FMS

V júli 2001 v práci [11] Fluhrer, Mantin, Shamir (odtiaľ názov útoku) ukázali, že algoritmus RC4 je slabý tým, že preň existujú semiačka, pri ktorých s istou pravdepodobnosťou niektorý bajt zo semiačka sa prenesie do prvého bajtu výstupného prúdu. Semiačko pre RC4 PRNG je zostrojené ako IV || kľúč (viď. obr. 4 1); také IV, ktoré dajú výstup odhaľujúci bajty kľúča, nazývame „slabé“. Na základe zachytených dvojíc {slabé IV, 1. bajt RC4 prúdu} je potom možné prehľadávaním podľa štatistického výskytu zistiť použitý tajný kľúč.

4.7.1 Slabé IV

Implementácia RC4 vo WEP má podľa FMS slabé IV:
K+3 | N-1 | X
kde K je poradie bajtu tajného kľúča (číslované od 0), ktorý môže byť týmto IV exponovaný, N je veľkosť stavového poľa, tu 256, a X je ľubovoľný bajt. Každé takéto IV má približne 5% šancu, že prenesie K-ty bajt tajného kľúča do prvého bajtu výstupu PRNG. Pri použití 64-bit WEP, ktorý má 40-bitový tajný kľúč sú slabé IV konkrétne (v hexadecimálnom tvare):
03FF??, 04FF??, 05FF??, 06FF??, 07FF??
Pri použití 128-bit WEP so 104-bitovým tajným kľúčom sú to:
03FF??, 04FF??, 05FF??, 06FF??, 07FF??, ..., 0FFF??

Neskôr (september 2003) Andrea Bittau v [12] dokázal pre FMS útok aj ďalšie, trochu zložitejšie skupiny slabých IV, ktoré majú približne 13% šancu na odhalenie jedného bajtu výstupu:

```
P | Q | K-2     pre P+Q=1 mod 256,  K ? {2, 3, 4, ... 12}
P | Q | K-1     pre P+Q=1 mod 256,  K = 0
P | Q | 254-K   pre P+Q=254-K mod 256, K ? {0, 2, 3, 4, ... 12}
```

Pre 128-bit WEP sú to:

```
ppqq00, ppqq01, ..., ppqq0A,    pre pp+qq = 01h mod 256
ppqqFF, pre pp+qq = 01h mod 256
ppqqF2, ppqqF3, ..., ppqqFC, ppqqFE,    pre pp+qq <= 0Ch mod FF
```

Spolu teda existuje 9472 slabých IV pre 128-bit WEP, čo je približne 0.0565% zo všetkých možných IV. Pre 64-bit WEP existuje 3328 slabých IV, čo je približne 0.0198% zo všetkých možných IV. Potrebujeme ich však zachytiť iba 5/13 z množstva potrebného pre prelomenie 128-bit WEP. Na prelomenie 64-bit WEP aj 128-bit WEP teda treba približne rovnaký počet rámcov.

4.7.2 Implementácia FMS útoku

Programy Airsnort, Aircrack-ng, a mnoho ďalších, umožňujú útok na WEP pomocou FMS metódy. Jedná sa o pasívny útok v monitorovacom režime, nie je ho teda možné spozorovať. Nazbieranie dostatočného množstva rámcov s rôznymi IV však môže na sieti s nízkou prevádzkou trvať niekoľko hodín, preto sa na urýchlenie môžu použiť aktívna reinjekcia rámcov alebo fragmentačný útok (opísané vyššie v 4.2 a 4.6).
Na úspešné zistenie šifrovacieho kľúča pomocou FMS útoku je potrebných niekoľko sto zachytených rámcov so slabým IV. Celkový počet šifrovaných rámcov potrebných na prelomenie WEP pomocou FMS je okolo 1 milióna, čo pri sieťach so silnou prevádzkou, alebo použitím agresívnej ARP reinjekcie je možné dosiahnuť na IEEE 802.11b aj g sieťach (dôležitá je odozva, nie prenosová rýchlosť) do 30 minút.

4.7.3 Obrana voči FMS útoku

Voči FMS útoku pri použití WEP boli pre vylepšenie štandardu IEEE 802.11 navrhnuté viaceré varianty:

• **Vylúčenie slabých IV** – všetky zariadenia musia používať výlučne „silné“ IV, jediné zariadenie na sieti, ktoré posiela slabé IV, kompromituje celú sieť. Niektorí výrobcovia toto na svojich zariadeniach implementovali, firma Agere Systems túto proprietárnu technológiu nazvala WEPplus (WEP+).
• **Vynechanie prvých 256 bajtov z výstupu PRGA** – toto riešenie bolo zavrhnuté z dôvodu nemožnosti implementácie na existujúcom hardvéri.

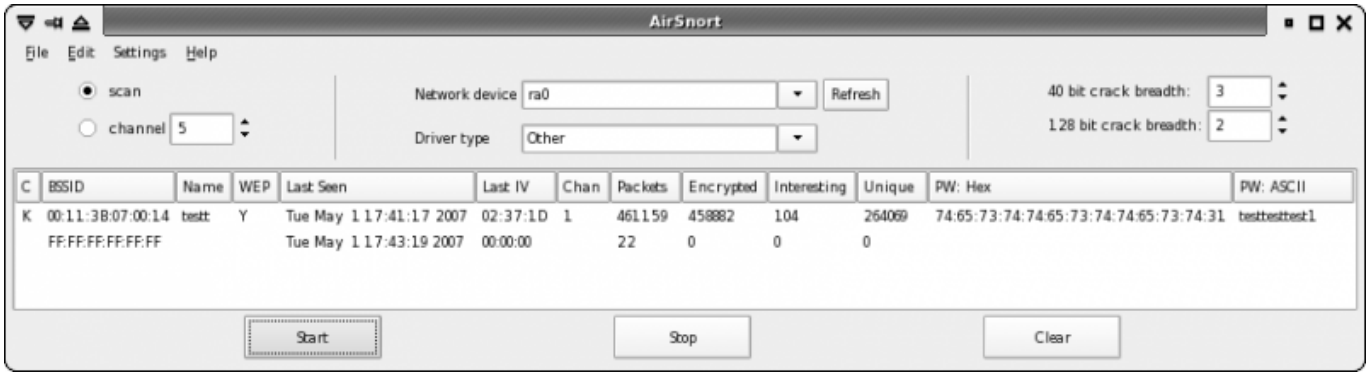
Riešenie, ktoré bolo štandardizované, je použitie RSN – TKIP na existujúcich zariadeniach, CCMP pre nové zariadenia (viď. 5 WPA a WPA2). V TKIP sú použité IV „silné“ a kľúč pre RC4 PRNG nie je statický, ale sa stále mení, preto FMS útok nie je možný.
Vylúčenie slabých IV je možné urobiť aj úpravou ovládačov ku sieťovej karte. Pre open-source ovládače to nie je ťažké, problém je zabezpečiť takúto „ručnú“ úpravu pre ovládače dodávané k OS Windows a pre firmware na AP. V ďalšej časti je však opísaný podobný útok, ktorý takúto snahu o proprietárne zabezpečenie ruší.

4.8 KoreK

V auguste 2004 na fóre netstumbler.org publikoval KoreK nový spôsob lámania RC4 algoritmu, a to zameraním sa nie na konkrétne hodnoty IV, ale na to, akým spôsobom je ovplyvnený Key Scheduling algoritmus (KSA). V práci [13] Rafik Chaabouni detailne popísal 17 KoreK útokov na KSA. Veľa z nich dáva falošné pozitíva (viac ako FMS), preto je nutné viac overovania dešifrovaním rámcov.

4.8.1 Implementácia KoreK útoku

Postupne od zverejnenia bol KoreK útok (čo je vlastne viacero KoreK útokov, ktoré „hlasujú“ o výsledku pre jednotlivé stavy KSA) implementovaný do všetkých programov, ktoré lámu WEP pomocou FMS. Airsnort verzia 0.2.7e (obr. 4 5) robí FMS a KoreK útoky paralelne a zobrazí výsledok z toho vlákna, ktoré ho dodá skôr. Aplikácia je intuitívna (stačí spustiť), použité sieťové karty bolo potrebné uviesť do monitorovacieho režimu manuálne.



obr. 4 5: Airsnort

4.8.2 Úspešnosť FMS/KoreK útoku

Pri všetkých zostrojených pokusoch sa úspešne podarilo nájsť šifrovací kľúč. Simulovaná prevádzka bola zameraná na maximalizáciu počtu paketov a teda nazbieraných IV, pomocou flood ping-u, ktorý na 11Mbit/s sieti (ad-hoc) generuje okolo 100 000 paketov za minútu (obojsmerne), na 54Mbit/s okolo 125 000 paketov za minútu. Keďže flood ping môže byť zneužitý na zahlcovanie, je nutné ho v OS GNU/Linux spustiť s root právami:

```
# ping 10.1.8.43 -f
```

KoreK útok je rovnako ako FMS pasívny, určený len na zistenie tajného kľúča. Ak by bola prevádzka na skutočnej sieti nízka, môžeme ju zvýšiť aktívnym útokom – reinjekciou alebo fragmentačným útokom, čím môžeme dosiahnuť takmer polovicu prevádzky simulovanej flood pingom (jednosmerne, v smere od útočníka idú totiž rámce s opakujúcimi sa IV).

celkový čas na prelomenie	paketov	šifrovaných	unikátnych IV	slabých IV pre FMS
2min 40sek	268351	266704	263955	61
2min 45sek	234186	232737	230238	351
2min 48sek	267629	265997	254014	145
2min 50sek	266771	265145	263094	111
2min 51sek	267981	266339	264032	62
2min 52sek	268439	266788	263905	116
2min 55sek	267660	266018	263364	147
2min 58sek	268390	266739	263969	49
3min 04sek	269276	267595	264007	602
3min 07sek	276817	275048	263854	98
3min 14sek	287278	285391	263907	299
4min 10sek	313459	310887	263746	339
5min 36sek	537704	534421	527743	57
5min 45sek	539790	536435	527727	97

tab. 4 1: Útoky na 64-bit WEP pomocou Airsnort na 11 Mbit/s ad-hoc sieti

celkový čas na prelomenie	paketov	šifrovaných	unikátnych IV	slabých IV pre FMS
2min 50sek	271407	269966	267497	110
2min 50sek	271616	270218	267539	618
2min 50sek	271634	270277	267418	297
3min 00sek	314183	312465	308796	328
3min 10sek	271271	269702	267545	47
3min 30sek	331438	329459	324545	55
5min 45sek	545589	542844	534703	60
5min 45sek	546294	543311	534514	150
5min 50sek	546171	543375	534954	41
5min 50sek	546504	544862	533998	346

tab. 4 2: Útoky na 128-bit WEP pomocou Airsnort na 11 Mbit/s ad-hoc sieti

celkový čas na prelomenie	paketov	šifrovaných	unikátnych IV	slabých IV pre FMS
2min 25sek	299211	297856	265375	352
2min 35sek	308054	306198	264498	131
2min 41sek	294213	292679	265335	111
2min 45sek	331572	329972	264872	743
2min 51sek	337028	335397	269117	345

tab. 4 3: Útoky na 128-bit WEP pomocou Airsnort na 54 Mbit/s infraštruktúrnej sieti

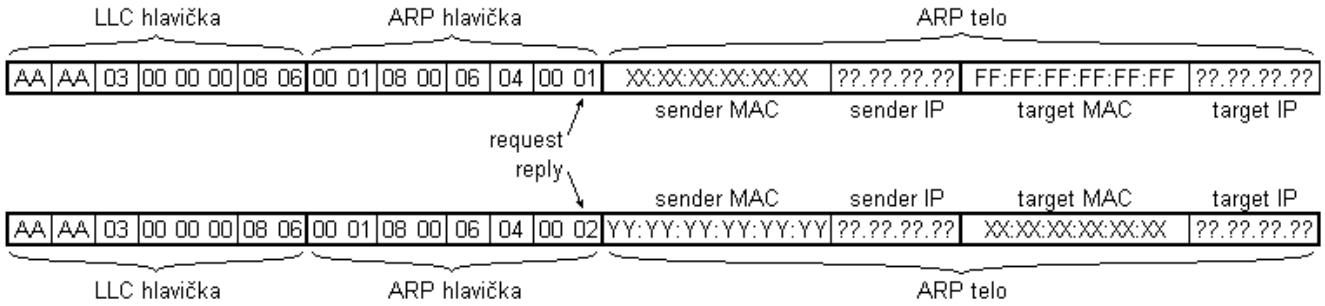
V tab. 4 1, tab. 4 2 a tab. 4 3 sú zapísané počty rámcov (Airsnort to nekorektne nazýva paketmi) potrebných na zistenie šifrovacieho kľúča pre viaceré pokusy. KoreK útok nerozlišuje silné a slabé IV ale pracuje s hodnotami v KSA, preto počet zachytených slabých IV nie je dôležitý údaj.
Vďaka silnej prevádzke sa podarilo kľúč zistiť vždy maximálne do 6 minút. Pri slabej prevádzke na sieti by tento útok trval niekoľko hodín, ak by sme však použili ARP reinjekciu (viď. 4.2.1) s rýchlosťou okolo 500 paketov/sek, vieme potrebných 500 tisíc rámcov nazbierať do 17 minút. Ak máme štastie, postačí na prelomenie 250 tisíc rámcov (viď. tabuľky vyššie), čo vďaka ARP reinjekcii nazbierame za menej ako 9 minút na sieti, ktorá mohla mať takmer nulovú prevádzku.

4.8.3 Obrana voči KoreK útoku

Pre KoreK útok neexistujú konkrétne hodnoty slabých IV tak ako pri FMS, nie je teda možné ich jednoducho vylúčiť.
Vhodnou obranou je použitie RSN (WPA/WPA2).

4.9 Kleinov útok

Andreas Klein na prednáške [14] v júni 2005 uviedol, a potom vo februári 2006 v [15] podrobne popísal nový druh útoku na RC4 šifru zameraný na celé stavové pole KSA. V apríli 2007 Erik Tews, Ralf-Philipp Weinmann, a Andrei Pyshkin v [16] popísali praktickú implementáciu tohoto útoku pre WEP a zverejnili proof-of-concept utilitu aircrack-ptw.



obr. 4.6: Prenášané dáta pre ARP request/reply rámce

Útok vyžaduje pre 128-bitové WEP poznať prvých 16 bajtov plaintextu, čo je možné pri ARP rámcoch (obr. 4 6). ARP request a reply sa líšia na 16. bajte, ale ľahko ich odlíšime podľa cieľovej MAC v IEEE 802.11 hlavičke, ktorá je nezašifrovaná – request je posielaný ako broadcast. Pomocou MAC adries vieme určiť alebo odhadnúť aj ďalšie bajty plaintextu (MAC adresa odosielateľa, IP adresa odosielateľa, MAC adresa cieľa, IP adresa cieľa), pre útok ale postačuje prvých 16.
Pri 40000 nazbieraných zašifrovaných ARP rámcoch vieme určiť tajný kľúč s pravdepodobnosťou 50%, pri 85000 ARP rámcoch s pravdepodobnosťou 95%.

4.9.1 Realizácia Kleinovho útoku

Kleinov útok je možné realizovať ako pasívny, ale ARP prevádzka na bežných sieťach je taká nízka, že by mohol trvať niekoľko dní. Preto je efektívnejšie realizovať ho ako aktívny útok, a to nasledovne:

1. monitorovať prevádzku pomocou wireshark alebo Aircrack-ng;
2. spustiť ARP reinjekciu pomocou aireplay-ng -3 (popísané v 4.2.1);
3. počkať na prirodzený ARP paket, alebo vynútiť si ho pomocou deautentifikácie (viď. 6.4);
4. počkať, kým sa reinjektuje dostatočné množstvo ARP paketov a uložiť zachytenú premávku do pcap súboru;
5. spracovať uložený pcap súbor pomocou aircrack-ptw:

```
$ ./aircrack-ptw many-arps.cap
This is aircrack-ptw 1.0.0
For more informations see <a href="http://www.cdc.informatik.tu-darmstadt.de/" title=
"http://www.cdc.informatik.tu-darmstadt.de/">http://www.cdc.informatik.tu-darmstadt.d
e/</a> aircrack-ptw/
allocating a new table
bssid = 00:11:3B:07:00:14  keyindex=0
stats for bssid 00:11:3B:07:00:14  keyindex=0 packets=25989
Found key with len 13: 74 65 73 74 74 65 73 74 74 65 73 74 31
```

Samotný výpočet trvá okolo sekundy na P4 2.6 GHz. Nazbierať 85000 ARP rámcov je možné do 3 minút pri 500 rámcoch/sek. V pokuse sa podarilo zistiť 104-bitový kľúč už pomocou 26000 rámcov, nazbieraných za 100 sekúnd. Práca [16] má už v názve lámanie WEP za menej ako 60 sekúnd, čo je

síce trochu zavádzajúce, ale je to možné realizovať.

4.9.2 Ochrana voči Kleinovmu útoku

Útoku je možné úplne zabrániť jedine zabránením posielania rámcov so známym začiatkom plaintextu. Čisto pasívne útoky môžeme minimalizovať pomocou statických tabuliek na všetkých staniciach, čo je ale ťažko manažovateľné a zle škálovateľné riešenie. Aktívny útočník však môže injektovať vlastný ARP rámec zostrojený pomocou RC4 prúdu získaného iným spôsobom (viď. 4.3 Zbieranie slovníka PRGA pomocou Shared-Key autentifikácie, 4.5 KoreK chopchop, 4.6 Fragmentačný útok). Takúto injekciu je možné obmedziť pomocou zahadzovania opakovaných IV (viď. 4.2.2 Ochrana voči reinjekcii) alebo útok odhaliť pomocou Wireless IDS.
Odporúčaným riešením je použitie RSN (WPA/WPA2), ktoré Kleinov útok znemožní.

URL článku: https://security-portal.cz/clanky/bezpe%C4%8Dnost-hacking-wifi-80211-3-wep

Odkazy:
[1] https://security-portal.cz/users/matej
[2] https://security-portal.cz/category/tagy/cracking
[3] https://security-portal.cz/category/tagy/hacking
[4] https://security-portal.cz/category/tagy/networks-protocols
[5] https://security-portal.cz/category/tagy/security
[6] https://security-portal.cz/category/tagy/wifi-wireless
[7] http://www.picocomputing.com/
[8] http://matej.sustr.sk/publ/dipl/
[9] http://creativecommons.org/licenses/by-nc/3.0/