TDDBC03 – Linköping 060507

# Praktisk WLAN-Säkerhet

801119-1973 Tony Sinisalu Lindgren
tonsi055
820604-9010 Johan Öhman
johoh683

## 1. Inledning

I dagens informationssamhälle är användandet av trådlösa nätverk relativt utbrett och ökar ständigt. Säkerheten i dessa är en viktig aspekt, i synnerhet för 802.11-nätverk som är standarden för trådlös kommunikation mellan datorer. Trafiken mellan datorer innehåller ofta känslig information som kreditkortsnummer, affärshemligheter o.s.v. Fysisk tillgång till nätverket krävs inte vilket gör trådlösa nätverk till betydligt intressantare mål för attackerare. Med en riktad antenn kan attack iscensättas upp till 2 km från målnätverket.

I det här projektet har vi avsett att undersöka vilka hot som finns mot 802.11-nätverk och hur man kan skydda sig mot dessa. Detta planerar vi att göra genom att ta reda på vilka färdigutvecklade attackverktyg som finns och använda dessa. Vårt mål är att få en ökad förståelse för hur säkerhet i trådlösa nätverk fungerar.

## 2. 802.11-Arkitektur

802.11-arkitekturen är uppbyggd av ett flertal olika komponenter och tjänster som samarbetar för att ge enheterna på nätverket möjlighet att kommunicera trådlöst. Enheterna utgörs av klienter och accesspunkter. Accesspunkterna fungerar som bryggor mellan trådlösa och trådbundna nätverk samt agerar som switch. Klienterna består ofta av bärbara datorer, PDA:er och stationära datorer, d.v.s. slutanvändare där nätverksapplikationer exekveras.

### 2.1. Nätverksparametrar

Vid installation av ett 802.11-nätverk finns det vissa primära parametrar som bör ses över vid konfiguration.

**2.1.1. SSID.** Identifiering av ett nätverk sker med hjälp av ett SSID (*Service Set IDentifier*). Detta bifogas alla paket som skickas i nätverket. Alla trådlösa enheter som försöker kommunicera med varandra måste ha samma SSID.

**2.1.2. BSSID.** BSSID (*Basic Service Set Identifier*) definieras av 802.11-standarden som MAC-adressen för accesspunkten i det BSS som den tillhör. Varje BSS definieras unikt av ett BSSID.

**2.1.3. Kanal.** Idag är de flesta nätverksenheter av modell 802.11b/g. Dessa använder sig av frekvenser mellan 2.4 och 2.485 GHz, vilket är ett olicensierat spektrum. Detta gör att andra anordningar såsom mikrovågsugnar och trådlösa telefoner kan använda sig av samma frekvenser vilket i vissa fall ställer till problem.

802.11-nätverk delar upp frekvensspektrumet i 11 kanaler och endast 3 kanaler (1,6 och 11) kan användas samtidigt utan att viss överlappning förekommer.

### 2.2. Drifthantering

802.11-standarden innehåller vissa tjänster för drifthantering och pålitlig kommunikation. Dessa är implementerade i alla 802.11-enheter, klienter såsom accesspunkter.

**2.2.1. Associering.** Associering skapar en logisk förbindelse mellan en klient och en accesspunkt. Klienter i ett trådlöst nätverk måste associeras med en accesspunkt innan den tillåts sända data genom denna. Efter associeringen tilldelas klienten ofta en IP-adress av en DHCP-server.

En klient kan vara associerad med endast en accesspunkt men en accesspunkt kan vara associerad med flera klienter.

**2.2.2. Disassociering.** Disassociering används dels av accesspunkter för att eliminera förbindelser med klienter, dels av klienter när de ska lämna nätverket. Efter en disassociering måste en klient återassocieras för att kunna kommunicera med accesspunkten igen.

**2.2.3. Autentisiering.** I 802.11-standarden finns autentisering implementerad för att kontrollera åtkomsten till nätverket. Autentiseringsmekanismen gör det möjligt för trådlösa enheter att identifiera varandra. Utan att blivit autentiserad tillåts inte en enhet kommunicera med resten av nätverket.

**2.2.4. Deautentisering.** Deautentisering används för att förhindra en tidigare autentiserad enhet att utnyttja nätverket. Deautentiseringen är oåterkallelig och kan inte avvärjas.

## 2.3. Topologi

Täckningen i ett trådlöst nätverk kallas BSS (*Basic Service* Set) och utgör grundstenen i ett 802.11-nätverk. BSS existerar i två grundformer, infrastrukturläge och ad-hoc-läge.

**2.3.1. Ad-hoc-läge.** Den enklaste formen av nätverks-topologi bildas av en samling klienter sammankopplade i ett icke-hierarkiskt nät. D.v.s. klienterna kommunicerar direkt med varandra utan hjälp av en basstation eller dylikt. Avsaknaden av central kommunikationsenhet hindrar klienter utan direktkontakt att kommunicera eftersom ingen form av vidarebefordran existerar. Kallas även för IBSS (*Independent Basic Service Set*).

**2.3.2. Infrastrukturläge.** En BSS i infrastrukturläge innehåller en accesspunkt som vidarebefordrar all trafik i nätverket. Till skillnad från ad-hoc-nätverk kommunicerar klienterna inte direkt utan genom accesspunkten. Accesspunkten är i de flesta fall förbunden med en hub, switch eller router som i sin tur är ansluten till Internet.

Då accesspunkter i två eller flera BSS:er är sammankopplade med någon form av distributionssystem skapas ett ESS (*Extended Service Set*). Detta möjliggör en viss grad av mobilitet för klienterna. [3], [10]

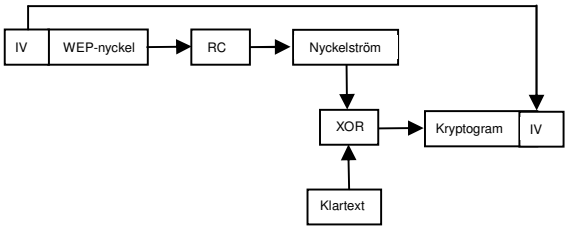
# 3. 802.11-säkerhet

De två mest använda säkerhetsmekanismerna i 802.11-nätverk idag är WEP (*Wired Equivalent Privacy*) och WPA (*Wi-Fi Protected Access*)

## 3.1. WEP

WEP är en del av IEEE 802.11-standarden och ett försök att för 802.11-nätverk uppnå ett skydd mot intrång jämförbart med det i trådbundna nätverk.

**3.1.1. Sekretess.** WEP använder sig av strömchiffret RC4 för att kryptera den klartext som varje pakets nyttolast består av. För att undvika att samma nyckelsekvens används upprepade gånger, vilket gör kryptotexten sårbar för kända klartextattacker, har varje paket en 24-bitars initieringsvektor. RC4 expanderar en sammanslagning av WEP-nyckeln och initieringsvektorn till en godtyckligt lång nyckelström som sedan kombineras med klartexten genom bitvis exklusiv eller (xor) vilket ger 224 olika nyckelströmmar. Vid dekryptering måste samma ström av nyckelbitar genereras och adderas modulo 2 (xor) till meddelandet. Alltså måste initieringsvektorn skickas okrypterad i varje paket. [1]




**Figur 1. Kryptering i WEP**

Återanvändning av initieringsvektorer utgör ett problem i WEP. Att den består av 24 bitar innebär att det efter endast ca 5000 paket förväntas förekomma kollisioner. Situationen förvärras då det inte existerar någon standard för hur initieringsvektorer ska hanteras. Vissa PCMCIA-kort återställer initieringsvektorn till noll varje gång de initialiseras, vilket sker varje gång de kopplas in i en bärbar dator. För varje paket ökas sedan initieringsvektorn med ett vilket gör att nyckelströmmar med låga värden på initieringsvektorer förekommer väldigt ofta när sådan hårdvara används.

Att initieringsvektorn och nyckeln används för att initiera RC4-shiffret innebär att repetition av nyckelströmmar sker när en initieringsvektor används upprepade gånger. Genom att addera två kryptotexter krypterade med samma nyckelström, modulo 2, erhålls de två klartexterna sammanslagna (exklusivt eller).

$C_1$ XOR $C_2$ = $P_1$ XOR RC4(iv, k) XOR $P_2$ XOR RC4(iv, k) = $P_1$ XOR $P_2$
C = kryptotext, P = klartext, k = nyckel, iv = initieringsvektor

Frekvensanalyser och andra kryptografiska hjälpmedel kan användas för att särskilja klartexterna. Flera delar av ett paket är förutsägbart vilket ger viss kännedom om klartexten. Att generera krypterad känd klartext över nätverket förenklar attacken avsevärt men kräver autentisering. När man känner klartexten är det lätt att beräkna nyckelströmmen som krypterat den. Eftersom det finns $2^{24}$ olika nyckströmmar (24 bitar initieringsvektor)

och ett pakets MTU (*Maximum Transmission Unit*) är ca 2300 byte kan en tabell på ca 36 GB innehålla alla möjliga nyckelströmmar.

Efter att ha beräknat n bitar av en nyckelström kan en induktiv klartextattack utföras. Välj en klartext som är 24 bitar kortare än den kända nyckelströmmen, beräkna kontrollsumman för denna och sänd n + 8 bitar till en accesspunkt. Om inget svar erhålls väljs ett av de andra 254 talen som de 8 extra bitarna kan representera tills ett svar fås. Detta ger kryptotexten motsvarande de sista 8 bitarna av kontrollsumman vilket betyder att ytterligare 8 bitar av nyckelströmmen kan beräknas. Upprepa tills nyckelström av önskad längs är funnen. [5], [18]

**3.1.2. Integritet.** När WEP-protokollet används beräknas för varje paket en kontrollsumma, som tillsammans med meddelandet utgör klartexten.


**Figur 2. Klartext i WEP**

Kontrollsumman är av typen CRC (*Cyclic Redundancy Check*) som baseras på division i en polynomring över en ändlig kropp med 2 element vilket ger den en linjär struktur. Alltså är kontrollsumman en linjär funktion av meddelandet vilket gör det möjligt att utföra kontrollerade ändringar utan att testet av kontrollsumman visar på avvikelser.

Antag att en attackerare har fångat upp ett paket som består av en initieringsvektor, *v* och det krypterade meddelandet, *C(m, c(m))*, där *m* är meddelandet och *c(m)* är kontrollsumman. Krypteringen i WEP består av exklusiv eller mellan klartexten och utdata från strömchiffret RC4 som initierats med den aktuella initieringsvektorn och WEP-nyckeln, *k*, vilket ger $C = RC4(v, k)\ XOR\ (m, c(m))$. Attackeraren vill nu byta ut *m* mot *m´* där $m´ = m + \Delta$ och fortfarande ge upphov till en korrekt kontrollsumma. För att lyckas med detta behöver denne bara beräkna $c(\Delta)$ och addera $(\Delta\ , c(\Delta))$ till båda sidor av ekvationen ovan. Efter några beräkningssteg skapas kryptogrammet, $C´ = RC4(v, k)\ XOR\ (m´, c(m´))$. Därefter kan attackeraren *spoofa* källan och därigenom vilseleda mottagaren. [1]

**3.1.3. Autentisering.** WEP använder en utmaning/respons-mekanism för att autentisera klienter till en accesspunkt. Klienten som ansöker om att bli autentiserad skickar en autentiseringsförfrågan som accesspunkten svarar på genom att sända ett slumpvis genererat, 128-bitar stort tal, en klartextutmaning. Klienten kopierar slumptalet till ett nytt paket som den krypterar med den delade WEP-nyckeln och skickar som svar till accesspunkten. Accesspunkten kontrollerar detta genom att själv kryptera klartexturmaningen och jämföra resultatet med klientens svar. Ett fjärde paket som innehåller resultatet av autentiseringen skickas som avslutning på autentiseringsprocessen. [5]

Autentiseringen existerar för att enheter i ett nätverk har behov av att på ett tillförlitligt sätt identifiera andra enheter. Tyvärr kan inte autentiseringsmekanismen i WEP tillfredställa detta behov. Kryptering i WEP sker som bekant genom bitvis addition modulo 2 mellan en nyckelström och en klartext. Eftersom autentiseringen innebär att samma klartext skickas både krypterad och okrypterad kan nyckelströmmen återskapas. Detta görs genom att addera den okrypterade klartexten, slumptalet, med kryptotexten av samma tal.

Följande beräkningar ger nyckelströmmen som skapats av RC4-skiffret med WEP-nyckeln och initieringsvektorn som indata.

```
C = K XOR R
=> C XOR R = K XOR R XOR R = K
R = slumptal, K = nyckelström, C = kryptogram
```

Att autentisera en enhet utan att ha tillgång till WEP-nyckeln är lätt om man beräknat nyckelströmmen. Genom att skicka en autentiseringsförfrågan förses man med ett nytt slumptal. Addition mellan det nya slumptalet och nyckelströmmen ger ett krypterat paket. Initieringsvektorn som användes vid den korrekta autentiseringen måste självklart bifogas paketet för att den autentiserande enheten ska beräkna en likadan nyckelström. När detta är gjort är WEP:s autentiseringsmekanism bruten. [7]

**3.1.4. Nyckelhantering.** Hur nyckelhanteringen i WEP ska skötas är inte entydigt bestämt. Tillverkare av hårdvara tillåts själva besluta hur deras enheter ska hantera WEP-nycklar. Två standardiserade metoder existerar dock. Den ena innebär att varje MAC-adress har en unik nyckel. Information om MAC-adresser och deras respektive nycklar finns tillgänglig i en tabell. I den andra metoden använder sig enheterna av en lista med fyra olika nycklar. För att veta vilken nyckel som använts skickas aktuellt nyckelindex okrypterat i varje paket. Enheterna kan alltså använda de fyra olika nycklarna för dekryptering. Kryptering av paket kan däremot bara ske med en på förhand bestämd nyckel.

Eftersom nycklarna vanligtvis bara kan ändras manuellt blir tiden de används ofta långvarig vilket är ytterligare en svaghet i WEP. [6], [7]

## 3.2. WPA

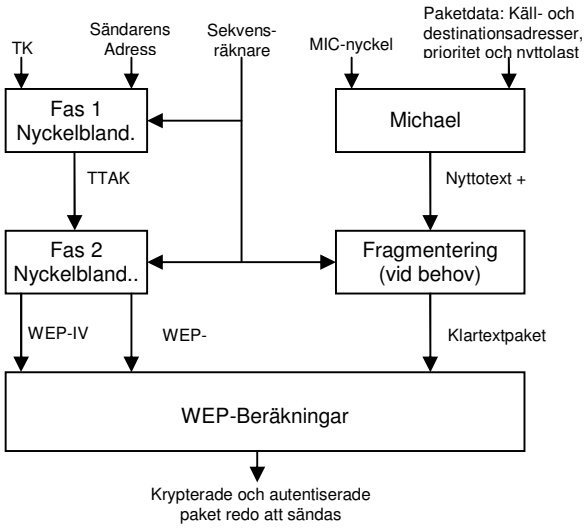
WPA (Wi-Fi Protected Access) utvecklades i väntan på standarden WPA2/IEEE 802.11i för att åtgärda flera av de allvarliga svagheter WEP lider av. En av de stora fördelarna med WPA om man inte tittar på säkerhetsaspekten är att det endast krävs en mjukvaruuppdatering. Om WPA krävt ny hårdvara för att införa en bättre säkerhetslösning, vilket 802.11i gör, hade lösningen blivit mycket dyrare och krångligare vilket förmodligen skulle ha resulterat i en begränsad spridning.

Den största svagheten i WEP var inte på grund av RC4 algoritmen utan grundades på skapandet av nyckeln och implementeringen av krypteringen. För att åtgärda detta används TKIP (Temporal Key Interchange Protocol) och autentisieringsmekanismen 802.1x. Detta tillför WPA dynamiskt nyckelutbyte, hierarkiskt nyckelsystem, skydd mot replay-attacker samt ömsesidig autentisiering. [11], [21]

**3.2.1. Sekretess.** För att adressera svagheterna med WEP används protokollet TKIP som ett lager av säkerhet runt en annars svag grad av kryptering. TKIP använder sig precis som WEP av strömchiffret RC4 för att kryptera klartexten som varje pakets nyttolast består av.

Initieringsvektorn som är förklarad i kapitlet om WEP är nu i WPA 48 bitar, det förhindrar att två paket med samma initieringsvektor kan fångas upp då det krävs närmare 900 år av kontinuerlig trafik innan en återanvändning sker. I WPA används initieringsvektorn som sekvensräknaren för TKIP. Sekvensräknaren är direkt kopplad till hur skapandet av nyckeln sker. Varje gång en ny nyckel skapas så nollställs sekvensräknaren. Varje paket som skickas räknar upp sekvensräknaren, TKIP ger ett skydd mot replay-attacker då paket med ogiltiga sekvensnummer försakas.

TKIP använder sig även av ett hierarkiskt nyckelhanteringssätt för att hantera hur attackerare utnyttjade den förutsägbarhet som statiska nycklar innebar. Varje paket som skickas är krypterat enligt WEP standarden, dock är nyckeln på 128 bitar och är unik för varje paket. Detta görs enligt en nyckelblandnings algoritm som kan ses i figur 3. För att prestanda inte ska bli lidande så har skapandet av nyckeln delats in i två faser. I fas 1 så skapas nyckeln utifrån sändarens MAC-adress, en temporär nyckel och de 32 högsta bitarna från sekvensräknaren. Den temporära nyckeln har skapats utifrån en huvudnyckel som både accesspunkt och klient kommit överens om, hur de kommer överens om en gemensam nyckel beskrivs i kapitlet om autentisiering. Fas 1 behöver bara räknas om en gång var 65 000: e paket. Utdatan som fås från fas 1 är sändarens adress och nyckel, dessa data används som inparameter i fas 2. I fas 2 blandas utdatan från fas 1 med de 16 låga bitarna från sekvensräknaren, detta ger en unik RC4 nyckel för varje paket. [10]




**Figur 3. Översikt av TKIP**

**3.2.2. Autentisiering.** Vid autentisiering i WPA används TKIP som sköter all nyckelhanteringsutbyte och 802.1x som är baserat på EAP (Extensible Authentication Protocol). EAP tillsammans med en RADIUS-server (Remote authentication dial-in user service) möjliggör ett flertal autentisieringsätt, vanligtvis använder man sig av en publik nyckel teknik för att göra en ömsesidig autentisiering.

Vid genomförd autentisiering förses klienten och autentisieringsservern med en MK (Master Key). Denna MK är en delad hemlighet mellan båda parterna, den används för att med hjälp av SSID:t på accesspunkten skapa en PMK (Pairwise Master Key). Autentisieringservern skickar PMK till accesspunkten, vilket innebär att klienten och accesspunkten har en gemensam nyckel. Med hjälp av en PMK kan nu nya temporära nycklar skapas, så kallade PTK (Pairwise Transient Key). Dessa PTK används för att kryptera data som skickas över den trådlösa länken.

För att möjliggöra fördelarna med WPA för hemmanätverk och mindre företag som inte har resurser för en RADIUS-server adderades möjligheten att autentisiera med ett känt lösenord, en så kallad PSK (Pre Shared Key). Det som är viktigt att skilja mellan de två olika autentisieringsmöjligheterna för WPA och WPA-PSK är endast hur de kommer överrens om huvudnyckeln som används för att skapa PMK, mekanismerna för att kryptera all data som skickas är densamma.

Då man i WPA-PSK använder sig av en förutbestämd nyckel som man själv ställer in har den vissa begränsningar. Nyckeln får bestå av mellan 8-63 tecken. För en någorlunda säkerhet bör minst 14 helt slumpvisa bokstäver och siffror användas, men man bör ha minst 20 för att maximera säkerheten. [10]

**3.2.3. Integritet.** Utöver de 32-bitarna CRC i WEP-protokollet utökades integritetsvalideringen i WPA med en MIC (Message Integrity Code) på 64-bitar kallad Michael, vilken är en del av TKIP protokollet. Detta ger avsevärt förbättrad integritet men olika typer av *bruteforce*-attacker är fortfarande möjliga. Därför finns en säkerhetsmekanism som när den detekterar ett felaktigt MIC-värde ställer om nycklarna för den kommunicerande stationen. Upptäcker den ytterligare ett paket med felaktigt MIC-värde inom en minut stänger den ner nätverket i 60 sekunder. Detta gör det möjligt att utföra en DoS attack i ett nätverk med WPA aktiverat, dock krävs det att paketen som skickas har giltiga sekvensnummer. [20]

### 3.3. WPA2/IEEE 802.11i

Precis som WPA förser WPA2 en höggradig säkerhet för all data som skickas samt att endast autentisierade användare har tillgång till det trådlösa nätverket. WPA2 har precis som WPA två möjliga sätt att autentisiera sig med, med IEEE 802.1x/EAP eller med en PSK.

Den stora skillnaden i WPA2 gentemot WPA är att all kryptering och meddelande-integritetskoll har ersatts av en enda komponent som anses vara helt säker, CCMP (*Counter Mode with Cipher Block Chaining Message Authentication Code Protocol*) vilket använder sig av AES (*Advanced Encryption Standard*) kryptering. I och med bytet till den starka krypteringsalgoritmen AES har svagheterna i RC4 krypteringen adresserats. Endast en 128 bitar lång nyckel används för att garantera en säker överföring.

Då AES kryptering kräver betydligt mer beräkningskraft än RC4 måste hårdvaran vara kompatibel för att användning av WPA2 ska vara möjlig. 802.11i/WPA2 inför den högsta grad av säkerhet vi kan uppnå idag för trådlösa nät, standarden anses vara framtidssäker. [12], [13]

## 4. Studie av diverse attacker mot WLAN

Nedan följer en teoretisk beskrivning av två attacker mot WLAN som är vanligt förekommande. Båda attackerna bygger på okunnighet hos användare och sårbarhet i operativsystemet.

### 4.1. Evil Twin

Vid en så kallad *evil twin*-attack sätter attackeraren upp en falsk accesspunkt i närheten av offret. Denna accesspunkt har en starkare signal än den legitima accesspunkten, detta för att få det ovetande offret att ansluta sig till den istället. För att försäkra sig om att signalen är starkare än den legitima accesspunkten kan en riktad antenn användas. Då attackeraren använder sig av samma SSID på sin accesspunkt som den legitima, märker offret ingen skillnad.

Attackeraren kan göra detta på flera olika sätt. Antingen kan attackeraren invänta nya användare som försöker ansluta sig till accesspunkten eller så kan denne disassiocera klienter. När de sedan försöker återansluta till Internet sker detta via den falska accesspunkten. Attackeraren kan då välja att ha en sidokanal, exempelvis attackerarens mobila bredband som offret kan ansluta igenom.

I det första fallet där attackeraren låter offret ansluta till Internet genom den falska accesspunkten kan attackeraren iscensätta MITM-attacker (*Man in The Middle*). Attackeraren kan observera, ändra och lägga till paket utan att de två kommunicerande parterna är medvetna om det.

I det andra fallet finns möjligheten att presentera en inloggningssida där denne fångar användarnamn samt lösenord och därefter ger ett felmeddelande som ber användaren återansluta. Återanslutningen för offret sker mot den legitima accesspunkten. Då allt gått så snabbt är det väldigt få användare som förstår att någon ohederlig handling har försiggått.

Dessa attacker är mer vanliga vid publika *hotspots* än hos företagsnätverk då de oftast använder skyddade och krypterade autentisieringsmekanismer. Hotet mot företagsnätverken uppstår när användarna använder publika WLAN för att ansluta till företaget och utsätter sig för *evil twin*-attacken. [19]

### 4.2. Rogue Accesspoints

En *rogue accesspoint* är en installerad, otillåten accesspunkt i ett nätverk, som nätverksadministratörerna inte har kännedom om. En otillåten accesspunkt kan innebära en säkerhetsrisk då de oftast inte följer säkerhetsprinciperna som används på den platsen där den är inkopplad. Det medför att den som försöker ansluta sig mot den otillåtna accesspunkten inte behöver autentisiera sig för att få tillgång till nätverksresurserna. En otillåten

accesspunkt kopplas in i en befintlig port till nätverket, det kan ske illvilligt eller av okunskap om trådlös säkerhet. [15]

## 5. Attacker i praktiken

Vi använde oss av Linuxdistributionen BackTrack som plattform för våra attacker. BackTrack är en sammanslagning av två ledande säkerhetsinriktade distributioner, Auditor och Whax, och innehåller förinstallerade attackverktyg.

För att utföra våra tester krävdes diverse hårdvara. Vi använde oss av en stationär dator med operativsystem Windows XP. Denna dator var trådlöst uppkopplad till en accesspunkt där både accesspunkten och nätverkskortet hanterade WEP och WPA. Som utgångspunkt för våra attacker användes en dator vars trådlösa nätverkskort hade stöd för WEP och WPA, nätverkskort var även tvunget att kunna agera som accesspunkt vid *fake portal*-attacken, samt att ha möjlighet att injicera paket för attacken mot WEP och WPA.

### 5.1. WEP-attack

WEP-attacken grundar sig på en attack kallad FMS som presenterades 2001 i en rapport skriven av Scott Fluhrer, Itsik Mantin och Adi Shamir. Rapporten visade att vissa, så kallade svaga, initieringsvektorer läcker information om WEP-nyckeln. Genom att fånga in tillräckligt många svaga initieringsvektorer kan nyckeln beräknas på relativt kort tid. Detta bygger dock på att attackeraren känner till inledningen av klartexten. Tyvärr inleds alla IP- och ARP-paket med 0xAA (SNAP-*Header*) vilket underlättar för attackeraren. [3],

**5.1.1. RC4.** FMS utnyttjar svagheter i strömchiffret RC4 och för att förstå hur WEP-nycklar beräknas i AirCrack krävs kunskap om hur RC4 fungerar. Två huvudalgoritmer KSA (*Key Scheduling Algoritm*) och PRGA (*Pseudo Random Generation Algoritm*) utgör huvudelen av RC4. Dessa presenteras nedan i form av pseudokod.

**5.1.1.1. KSA.** KSA-algoritmen skapar en tillståndsarray med värden mellan 0 och 255 som den sedan slumpvis förflyttar inom arrayen.

```
KSA(K)
K[] = Secret Key array
Initialization:
For i = 0 to N - 1
S[i] = i
j = 0
Scrambling:
For i = 0 to N - 1
j = j + S[i] + K[i mod l] // l = length of
Secret Key Array
Swap(S[i], S[j])
```

**5.1.1.2. PRGA.** Denna algoritm använder sig av tillståndsarrayen skapad i KSA och genererar nyckelströmmen som senare adderas till klartexten. [3]

```
PRGA(K)
Initialization:
i = 0
j = 0
Generation Loop:
i = i + 1
j = j + S[i]
Swap(S[i], S[j])
Output z = S[S[i] + S[j]]
```

**5.1.2. Beräkning av WEP-nyckel i AirCrack.** Följande genomgång av hur WEP-nyckeln beräknas i AirCrack är tänkt att fungera som ett belysande exempel och kommer från ref. 8

Den svaga initieringsvektorn som används är (3, 255, 7). Alltså ser nyckelarrayen ut som följer K[0] = 3, K[1] = 255, K[2] = 7 K[3 – 7] = ? (40-bitars nyckel)

RC4 inleds med skapandet av tillståndsarrayen i KSA. Första steget i denna ser ut som följer:

Steg 1:
```
i = 0, j = 0,  S[0] = 0, S[1] = 1, S[2] = 2
o.s.v.
=> j = j + S[i] + K[i mod 8] = 0 + 0 + 3 = 3
Swap (S[i], S[j]) = Swap(S[0], S[3]) => S[0] =
3, S[3] = 0
```

Steg 2:
```
i = 1, j = 3, S[0] = 3, S[1] = 1, S[2] = 2, S[3]
= 0
j = j + S[i] + K[i mod 8] = 3 + 1 + 255 = 259
mod 256 = 3
Swap (S[i], S[j]) = Swap(S[1], S[3]) => S[1] =
0, S[3] = 1
```

Steg 3:
```
i = 2, j = 3, S[0] = 3, S[1] = 0, S[2] = 2, S[3]
= 1
j = j + S[i] + K[i mod 8] = 3 + 2 + 7 = 12
Swap (S[i], S[j]) = Swap(S[2], S[12]) => S[2] =
12, S[12] = 2
```

Steg 4: (Hittills har endast kända värden på nyckeln använts men i steg 4 är nyckelbitarna okända.)
```
i = 3, j = 12, S[0] = 3, S[1] = 0, S[2] = 12,
S[3] = 1, S[12] = 2
j = j + S[i] + K[i mod 8] = 12 + 1 + ? = ?
Swap (S[i], S[j]) = Swap(S[3], S[?]) => S[3] =
??, S[?] = 1
```

Eftersom den första delen av klartexten är känd som AA hexadecimalt är det enkelt att beräkna den inledande nyckelströmmen genom addition modulo 2 mellan de första 16 bitarna kryptogram och den kända klartexten. Antag att denna kryptotext har värdet A5 hexadecimalt. Detta ger klartexten F hexadecimalt, AA XOR A5 = F.

Att beräkna det okända värdet på j i det fjärde steget av KSA är nu möjligt genom att undersöka hur PRGA genererar nyckelströmmen.

Under inledningen av genereringsloopen sätts i till 1, `i = i + 1 = 0 + 1 = 1` och j till 0, `j = j + S[i] = 0 + 0 = 0`. Efter detta byts värdet på `S[1]` och `S[0]` vilket ger `S[1] = 3` och `S[0] = 0`. Dessa värden adderas därefter till ett index för tillståndsarrayen. Värdet i tillståndsarrayen för detta index blir utdata.

```
z = S[S[i] + S[j]] = S[S[1] + S[0]] = S[0 + 3] = S[3] = ?
```

Värdet på S[3] motsvarar inledningen på nyckelströmmen som har värdet 0xF, 15 decimalt vilket kommer att vara värdet på S[3] efter steg 4 av KSA.

Nu kan steg 4 av KSA algoritmen slutföras:

```
Swap (S[i], S[j]) = Swap(S[3],S[15]) => S[3] = 15, S[15] = 1
```

Alltså är värdet på j = 15 vilket ger :

```
15 = 12 + 1 + ? <=> ? = 15 – 12 – 1 = 2
```

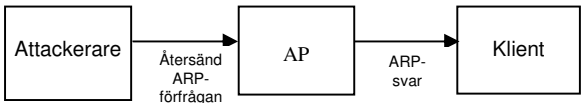
Värdet på `K[3]` är alltså 2. [8]

Eftersom KSA-delen av RC4 körs klart innan PRGA-delen genererar nyckelströmmen kommer det beräknade värdet på `K[3]` i de flesta fall inte att stämma. Däremot kommer KSA, fem procent av gångerna, inte att ändra värdena i `S[0] – S[3]` efter de tre inledande permutationerna vilket gör att det beräknade värde på `K[3]` är korrekt. AirCrack måste alltså utföra beräkningarna med ett flertal svaga initieringsvektorer och statistiskt bestämma det korrekta värdet.

**5.1.3. Attackverktyg.** När vi utförde attacken använde vi oss av några, i BackTrack, förinstallerade program distribuerade under GNU GPL.

**5.1.3.1. Kismet.** För att identifiera lämpliga nät att attackera, bestämma SSID, BSSID, kanal, o.s.v., använde vi oss av nätverksidentifieraren Kismet. Kismet kan detektera både accesspunkter och klienter samt se vilka associationer som finns på nätverket. Programmet arbetar dessutom passivt vilket är att föredra eftersom inga paket då kan loggas.

**5.1.3.2. Aireplay.** När vi hittade ett lämpligt nätverk, vårt testnätverk, visste vi att vi behövde minst 500 000 unika initieringsvektorer för att bestämma en 128-bitars WEP-nyckel. Eftersom mängden krypterad trafik hos det aktuella nätverket var för liten skulle detta ta väldigt lång tid. ARP (*Adress Resolution Protocoll*) används för att översätta IP-adresser till MAC-adresser. Genom att fånga upp ARP-förfrågningar och återsända till accesspunkten genereras krypterad trafik vilket ökar antalet initieringsvektorer som skickas i nätverket avsevärt.




**Figur 4. Injiciering av ARP-paket**

Att först deautentisera klienten på nätverket ökar chansn att hitta giltiga ARP-paket. Detta åstadkom vi med paketinjiceringsprogrammet Aireplay.

**5.1.3.3. Airodump.** Innan vi inledde injiceringen av ARP-förfrågningar startade vi Airodump för att logga den krypterade trafiken över nätverket. Det vi är ute efter är ju egentligen initieringsvektorer så Airodump ställdes in för att i huvudsak fånga in dessa.

**5.1.3.4. Aircrack.** När vi hade fångat in ca 500 000 initieringsvektorer startade vi verktyget Aircrack som med hjälp av den infångade informationen beräknade WEP-nyckeln

## 5.2. WPA-attack

En WPA-attack skiljer sig mot en attack mot WEP. Oberoende på hur bra eller dåligt man har valt lösenordet till en WEP krypterad uppkoppling så kan det knäckas, allt beror på hur många initieringsvektorer man fångar. Lösenordet i WPA kan göras tillräckligt bra för att göra en attack praktiskt taget omöjligt. Attacken mot WPA kan även ses vara lite mer attackvänlig än den mot WEP, då man endast behöver vara i närheten av accesspunkten en kort begränsad tid för att fånga upp korrekt paket. Sedan kan återstående delen av attacken göras offline.

Huruvida en attack har en möjlighet att lyckas beror på om autentisieringen för accesspunkten sköts av en RADIUS-server eller om en PSK används. Används en RADIUS-server är det i princip omöjligt att utföra en lyckad attack. Beroende på om nätverksadministratören för accesspunkten har valt ett enkelt lösenord som kan utsättas för en *bruteforce*-attack är attacken mot WPA-PSK möjlig.

Som tidigare nämnts utnyttjas inte någon svaghet i TKIP, då det inte ses som möjligt att attackera. I synnerhet som varje paket får en ny nyckel. Det är svagheter i initialiseringsprocessen av WPA-PSK som utnyttjas i attacken. [14]

**5.2.1. Initialiseringsprocess.** En PSK är ett 256 bitar långt nummer eller ett lösenord mellan 8-63 tecken.

Accesspunkten har en PSK som är bunden till dess SSID, denna kallas PMK. Det är med hjälp av en PMK som 4-vägshandskakningen startas och skapar den temporära nyckeln PTK.

Skapandet av en PMK görs enligt följande formel:

```
PMK = PBKDF2(lösenord, ssid, ssidLängd, 4096, 256)
```

Funktionen PBKDF2 länkar samman lösenordet (PSK), SSID:t, längden på SSID:t och sedan hashar den 4096 gånger för att generera ett 256 bits värde för att få fram PMK.

Den temporära nyckeln PTK skapas genom att man använder hashfunktionen HMAC (keyed-hash message authentication code) på PMK, MAC-adressen från klienten och accesspunkten samt de två nonce-värdena från de två första EAPoL (Extended Authentication Protocol over LAN) meddelandena i 4-vägshandskakningen. Där ett nonce-värde är slumpmässigt värde som används endast en gång.

**5.2.2. Lösenords-attack mot WPA-PSK.** För att lyckas med en attack mot WPA-PSK måste en giltig PTK kunna genereras. Som beskrevs i kapitlet om initialiseringsprocessen för WPA så behöver vi två MAC-adresser, två nonce-värden och ett korrekt lösenord för att generera en PTK. Vi saknar dock ett korrekt lösenord. Dessa värden kan fångas upp från de fyra meddelandena i 4-vägshandskakningen. För att fånga upp handskakningen kan det vara smidigt att disassociera en klient och vänta på dess åter-association. [16]

Tillvägagångssättet är som följer:

1. Fånga de fyra EAPoL meddelanden som utbyts under 4-vägshandskakningen mellan klienten och accesspunkten.
2. Konvertera ett ord från en lösenordslista till en PMK.
3. Generera en PTK med hjälp av PMK i steg 2 samt MAC-adresserna från klienten och accesspunkten och de två nonce-värdena som fås från meddelandena i steg 1.
4. Jämför MIC värdet från den genererade PTK med MIC värdet från det korrekta som kan återfinnas i det fjärde EAPoL meddelandet. Om det överensstämmer så har man funnit korrekt lösenord, annars får man börja om med steg 2 med ett nytt lösenord.

**5.2.3. Utförande.** Att utföra en lösenordsattack mot WPA-PSK är en förhållandevis enkel procedur, särskilt med tanke på att alla verktyg är färdigutvecklade och redo för användning. Även i denna praktiska attack förutsätts det att linuxdistributionen BackTrack används och att WLAN-kortet är korrekt uppsatt i mode *monitor*.

Då denna attack är inriktad mot att försöka hitta rätt lösenord så kommer även en stor textfil bestående av potentiella lösenord att behövas. Filen vi använde oss av var en sammanslagning av Websters engelska ordbok och flera listor av vanliga lösenord. Den totala summan av ord hamnade på ca 172 000 ord.

**5.2.3.1. Kismet.** Vi använde oss av Kismet för att identifiera en AP som använde sig av WPA samt anteckna SSID och MAC-adress för vår mål-accesspunkt.

**5.2.3.2. Airodump.** Med hjälp av det här programmet kan man spara ner paket i en fil. Ethereal kunde även ha använts men av någon anledning så fungerade det inte för oss. En annan smidig funktion med Airodump var att vi enkelt kunde se information om vilka klienter som var anslutna till vår mål-accesspunkt.

**5.2.3.3. Aireplay.** Då vi vill upprepa initialiserings-processen hos WPA så måste vi deautentisiera en klient. Detta görs enkelt med Aireplay då det har en inbyggd funktion för att skapa ett deautentisieringsmeddelande. I syntaxen måste vi skriva in MAC adressen till både AP och klienten, dessa har vi fått fram med hjälp av Kismet och Aircap. Efter att klienten kopplat upp sig igen så avbröt vi Aircap.

**5.2.3.4. Ethereal.** Att sniffa och analysera paket görs enkelt i Ethereals grafiska gränssnitt. Vi öppnade filen vi sparade med hjälp av Aircap och filtrerade bort allt utom EAPoL-paket då vi var intresserade av 4-vägshandskakningen. Dessa paket sparades ner i en fil för framtida användning.

**5.2.3.5. cowpatty.** För bestämma den förutbestämda nyckeln används programmet coWPAtty. Vad den gör är att med hjälp av en lösenordsfil, nätverkets SSID och EAPoL-meddelandena som utbyttes mellan accesspunkten och klienten kan testa att generera en temporär nyckel PTK utifrån en huvudnyckel (PMK). När man skapar den temporära nyckeln används MAC-adresserna på de inblandade stationer samt två nonce-värden. Om skapade paketets MIC och det verkliga överensstämmer så vet man att man har hittat rätt lösenord.

Vi använde oss av vår lösenordsfil, SSID på accesspunkten och filen vi sparade från Ethereal. Ca 152 sekunder senare hade vi hittat det 8 bokstäver långa lösenordet, cowPatty testade ca 42.57 nycklar/sek.

**5.2.3.6. Genpmk.** Att för coWPAtty konvertera ett ord med hjälp av ett SSID till en huvudnyckel (PMK) är en långsam process. Därför skapades programmet genpmk för att skapa en tabell bestående av färdiga hashvärden. Att skapa en tabell för alla SSID skulle ta en oerhörd plats så vi skapade en hashtabell endast för vår mål-accesspunkt. En färdighashad tabell, på totalt 8.5 GB, uppbyggd av 172 000 lösenord och 1000 vanliga SSID finns att ladda ner på www.churchofwifi.org.

Vi testade att köra coWPAtty en gång till, fast denna gång med en hashad tabell istället för en lösenordstabell i klartext. Detta test gjordes för att se vilken prestandaökning vi skulle kunna få i vår *bruteforce*-attack. Lösenordet hittades även denna gång fast nu efter redan 0.15 sekunder! cowPatty kunde nu testa 42792 nycklar/sek.

## 5.3. Fake Portal

Användandet av trådlösa *hotspots* är nuförtiden ett relativt utbrett fenomen. Att som användare koppla upp sig mot dessa är väldigt enkelt. Efter att man skapat ett konto och valt betalningsmetod kommer man till en inloggningssida där man anger användarnamn och lösenord. Tyvärr gör detta det enkelt för attackerare att få tillgång till känslig information som kreditkortsnummer, inloggningsuppgifter o.s.v.

Vad som behöver göras är i huvudsak att få användarna att koppla upp sig mot en falsk accesspunkt. Eftersom windowsdatorer väljer den accesspunkt med bäst signal räcker det ibland att den falska accesspunkten befinner sig nära måldatorn men för att vara på den säkra sidan kan man samtidigt utför en DoS-attack mot den riktiga accesspunkten.

Genom att visa en inloggningssida, liknande den korrekta, lurar man användare att ge ut sina inloggningsuppgifter. Om man dessutom vidarebefordra all trafik kan man utföra man-in-the-middle-attacker och på så sätt göra än mer skada.

**5.3.1. Airsnarf.** Verktyget Airsnarf är skrivet av ”the shmoo group” och används för att skapa falska trådlösa accesspunkter.
(Mer text tillkommer när vi är klara med attacken.)

# 6. Avslutning

Det här projektet gjordes med syfte att undersöka hur säker en vanlig användare är mot virtuella hot och angrepp. Efter en genomförd studie kan vi konstatera att den typiska användaren är väldigt sårbar. Säkerheten i dagens trådlösa nätverk behöver ses över och information om detta ämne måste spridas. Vid ett test i vårt bostadshus gjordes en avsökning för att se vilka tillgängliga trådlösa nät som vi kunde få kontakt med. Vi hittade sju stycken nät, två helt öppna, fyra stycken med WEP-kryptering och endast en med WPA2. Som denna rapport visar kan en WEP-kryptering forceras på ett fåtal minuter. Detta kan ses som ett bevis för att även bland högutbildade studenter där IT-miljö är en del av vardagen har inte budskapet om de omfattande säkerhetsluckorna nått fram. Det som även kan sägas är att den studenten som använde sig av WPA2 som utnyttjar en höggradig AES-kryptering, så det är inte helt omöjligt att det var en av våra kurskamrater eller någon om läst en liknande kurs tidigare.

Alla hot mot ett trådlöst nätverk går inte att skydda sig mot, men man kan undvika de flesta. Efter att skrivit denna rapport har vi fått en djupare inblick i säkerhetsmekanismerna man kan använda och dess svagheter. Genom att utföra dessa praktiska tester kommer vi även i framtiden att lättare kunna identifiera och förhindra en attack mot vårt nät.

Avslutningsvis vill vi med denna rapport förmedla att om möjligt, använd ett trådbaserat nät. Insisterar ni på ett trådlöst nät, använd minst WPA med ett lösenord på 20 helt slumpmässiga bokstäver och siffror och hoppas att ni inte blir utsatt för en mer sofistikerad attack.

## Källor


[18] W. Arbaugh, *An Inductive Chosen Plaintext Attack against WEP/WEP2*, http://www.cs.umd.edu/~waa/attack /v3dcmnt.htm, 2006-05-04

[5] W. Arbaugh, N. Shankar, J. Wan, *Your 802.11 Wireless Network has No Clothes*, 2001.

[10] Avaya Communicatoins, *802.11 Architecture*. http:// wireless.ictp.trieste.it/school_2002/lectures/ermanno/HTML/ /802.11_Architecture.pdf, 2006-05-03

[1] N. Borisov, I. Goldberg, D. Wagner. *Intercepting Mobile Communications: The Insecurity of 802.11*, 2001

[4] S. Fluhrer, I. Mantin, A. Shamir, *Weaknesses in the Key Scheduling Algorithm of RC4*, 2001.

[20] S. Fogie, *Cracking Wi-Fi Protected Access (WPA), Part 1*, http://www.informit.com/articles/article.asp?p=369221&seqNum=3, 2006-04-25

[14] S. Fogie, *Cracking Wi-Fi Protected Access (WPA), Part 2*, http://www.informit.com/articles/article.asp?p=370636&seqNum=1, 2006-04-27

[15] J. Geier, *Identifying Rogue Accesspoints*, http://www.wi-fiplanet.com/tutorials/article.php/1564431, 2006-05-04

[21] J. Geier, *WPA Security Enhancements*, http://www.wi-fiplanet.com/tutorials/article.php/2148721, 2006-04-29

[17] C. He, J. Mitchell, *Analysis of the 802.11i 4-Way Handshake*, 2004

[11] T. Higgins, *Wi-Fi Protected Access (WPA) NeedToKnow*, http://www.tomsnetworking.com/2003/06/25/wi/, 2006-04-28

[12] T. Higgins, *WPA - Wireless Security for the rest of us*, http://www.tomsnetworking.com/2002/11/01/wpa_/, 2006-04-27

[7] M. Kim. *Investigation about the RC4 and the weakness of WEP*, 2004.

[9]. J. Kurose, K. Ross, *Computer Networking. - A Top Down Approach Featuring the Internet,* Pearson Education, 2005.

[6] T. Newsham, *Cracking WEP Keys – Applying Known Techniques to WEP keys*, 2001

[16] F. Olsson, *Säkerhet i trådlösa nätverk*, Datormagazin, nr 3, 2006

[8] C. Peikari, S. Fogie*, Cracking WEP,* 2003

[19] L. Phifer, *Defeating Evil Twin attacks*, http://searchsecurity.techtarget.com/general/ 0,295582,sid14_gci1167674,00.html?track=wsland, 2006-04-27

[3] M. Roman, L. Fallet, S. Chandel, N. Nassif, *Reverse Engineering of AirCrack Software*, 2005.

[2] A. Vladimirov, K. Gavrilenko, A. Mikhailovsky. *Wi-Foo – The Secrets of Wireless Hacking*, 2004.

[13] Wi-Fi Alliance, *Deploying Wi-Fi Protected Access (WPA) and WPA2 in the Enterprise*, 2005