VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ
DEPARTMENT OF COMPUTER SYSTEMS

AUTOMATIZOVANÉ ÚTOKY NA WIFI SÍTĚ S NÍZKOU DETEKOVATELNOSTÍ A OBRANA PROTI NIM

AUTOMATED ATTACKS ON WIFI NETWORKS WITH LOW DETECTABILITY AND DEFENSE AGAINST THEM

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **JAN KLUSÁČEK**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. MICHAL ORSÁK**
SUPERVISOR

BRNO 2021

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav počítačových systémů (UPSY) Akademický rok 2020/2021

Zadání bakalářské práce

Student: **Klusáček Jan**
Program: Informační technologie
Název: **Automatizované útoky na WiFi sítě s nízkou detekovatelností a obrana proti nim**
Automated Attacks on WiFi Networks with Low Detectability and Defense Against Them
Kategorie: Bezpečnost

Zadání:

1. Seznamte se s některou z linuxových distribucí určenou pro penetrační testování a s problematikou prolamování běžných zabezpečení WiFi sítí.
2. Experimentujte s existujícími nástroji pro prolamování zabezpečení těchto sítí a se zranitelnostmi konkrétních přístupových bodů AP.
3. Vyhodnoťte možnosti detekování útoků při použití těchto nástrojů a popište omezení daných nástrojů.
4. Z vybraných nástrojů sestavte nový nástroj pro automatizované prolamování WiFi sítí.
5. Vyhodnoťte účinnost a detekovatelnost vašeho nástroje v závislosti na konfiguraci sítě. Diskutujte možnosti ochrany WiFi sítí před vytvořeným nástrojem.
6. Zhodnoťte dosažené výsledky.

Literatura:

- Dle pokynů vedoucího.

Pro udělení zápočtu za první semestr je požadováno:

- Splnění bodů 1 a 2 zadání.
- Seznam použitelných nástrojů pro bod 4.

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce: **Orsák Michal, Ing.**
Vedoucí ústavu: Sekanina Lukáš, prof. Ing., Ph.D.
Datum zadání: 1. listopadu 2020
Datum odevzdání: 12. května 2021
Datum schválení: 30. října 2020

Abstrakt

Cílem této bakalářské práce je sestavení nástroje pro automatizované prolamování Wi-Fi sítí. V této práci se čtenář dozví, jaká existují zabezpečení Wi-Fi sítí a jaké útoky je možné na konkrétní zabezpečení provádět. Dále jsou popsány konkrétní nástroje využívané pro tyto útoky, ze kterých jsou poté některé vybrány a zařazeny do výsledného nástroje.

Abstract

The goal of this bachelor's thesis is to create a tool which can be used for automated Wi-Fi attacks. All types of Wi-Fi security are described, with attacks, which can be used on specific security type. Existing tools used for hacking are described further, of which several are included to the final tool.

Klíčová slova

Wi-Fi, bezdrátové sítě, legální odposlechy, zabezpečení, prolamování, aircrack-ng, hashcat, WEP, WPA, WPA2, slovníkový útok

Keywords

Wi-Fi, wireless networks, legal interceptions, security, cracking, aircrack-ng, hashcat, WEP, WPA, WPA2, dictionary attack

Citace

KLUSÁČEK, Jan. *Automatizované útoky na WiFi sítě s nízkou detekovatelností a obrana proti nim*. Brno, 2021. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Michal Orsák

Automatizované útoky na WiFi sítě s nízkou detekovatelností a obrana proti nim

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Michala Orsáka. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Jan Klusáček
7. května 2021

Poděkování

Rád bych poděkoval vedoucímu této práce Ing. Michalu Orsákovi za cenné rady a pomoc při řešení problémů, na které jsem narazil.

Obsah

Kapitola 1

Úvod

V době vzniku této práce jsou Wi-Fi sítě hojně využívány v domácnostech, veřejných i firemních prostředích. Rozšiřování Wi-Fi sítí je spojeno s masovým rozšířením chytrých telefonů a dalších přenosných forem výpočetní techniky. Díky své propustnosti, která začíná přesahovat 1Gb/s, a jednoduché instalaci do požadovaných prostor, bezdrátové sítě postupně nahrazují i standardní drátové přípojky, protože je pro uživatele použití bezdrátového řešení mnohdy jednodušší a dostatečné. Použití bezdrátového řešení také umožňuje volný pohyb uživatelů a odpadá nutnost manuálního připojování, což je pro zařízení typu chytrý telefon esenciální. Uživatelé jsou tak limitováni pouze dosahem dané Wi-Fi sítě.

Jelikož ale internet může sloužit i jako místo pro vyjednávání nelegálních záležitostí a k různým nezákonným aktivitám, je občas nutné zjistit, k čemu daní uživatelé internet používají. Z toho důvodu vznikla potřeba vytvořit nástroj, který by umožňoval odposlouchávat vybrané sítě a zjišťovat z nich citlivé informace, k jejichž výměně na síti mezi uživateli dochází.

Téměř veškeré Wi-Fi sítě mají nějaké zabezpečení, jehož cílem je zabránění právě tomuto odposlechu. U Wi-Fi sítí existují různé typy zabezpečení, které jsou popsány v kapitole 2. Obecně platí, že starší zabezpečení mají více zranitelností a dají se tak jednodušeji prolomit. Novější zabezpečení se snaží tyto zranitelnosti opravovat, i tak má ale v podstatě každé z nich určitou slabinu, díky níž je možné ho prolomit. S pomocí získaného hesla je pak možné provoz na dané síti pomocí dalších nástrojů dešifrovat a odposlouchávat. Nástrojů, které umožňují prolomení zabezpečení, existuje mnoho, některé z nich jsou popsány v kapitole 4. Nicméně ne všechny jsou jednoduché na použití, a ne všechny jsou funkční.

Cílem této práce je tedy nalézt nástroje, které jsou aktuální, a pomocí nich vytvořit jeden nástroj, který bude sloužit k prolamování hesel Wi-Fi sítí se zabezpečením WEP a WPA/WPA2, a bude jednoduché ho použít. Zároveň by jeho činnost měla být co nejméně odhalitelná, aby uživatelé dané sítě neměli podezření, že je na síť prováděn útok. Možnosti detekování útoků a obrany proti nim jsou shrnuty v kapitole 5, samotné implementaci nástroje se věnuje kapitola 7.

Tento nástroj byl vytvářen v rámci projektu FlexProbe, jehož cílem je vytvořit flexibilní sondu pro zákonné odposlechy, kterou by mohly použít orgány činné v trestním řízení pro bezztrátový záchyt síťové komunikace v koncových sítích s rychlostí linek 10 Gb a ve Wi-Fi sítích. Sonda vzniklá v rámci tohoto projektu je složena z prefiltru implementovaném v FPGA. Filtrování je založené na technikách aproximate computingu. Obslužný software pak provádí dofiltraci provozu a exportuje odposlouchávaná data na cílové úložné zařízení.

Kapitola 2

Analýza zabezpečení

Soubor IEEE 802.11x standardů [5] označován jako Wi-Fi dává dohromady bezdrátovou síťovou technologii, která poskytuje přístup k internetu pomocí rádiových vln. Její hlavní výhodou je, že zařízení v této síti nemusí být spojeny pomocí kabelů. Stačí, aby daná zařízení byla vybavena bezdrátovým síťovým adaptérem a mohla tak komunikovat s přístupovým bodem. Přístupový bod, který bychom mohli nazvat jako základní stavební kámen Wi-Fi sítí, poskytuje bezdrátový signál, který mohou zařízení detekovat a následně pomocí něj navázat připojení k síti. V této kapitole jsou popsány protokoly zabezpečení Wi-Fi.

Aby se zařízení mohlo připojit k přístupovému bodu, je nutné, aby se nejdříve se sítí asociovalo. Tento proces probíhá v následujících krocích:

1. Zařízení vyšle *Probe request* - požadavek na zjištění dostupných sítí
2. Přístupový bod, který obdrží tento požadavek, na něj odpoví
3. Následně zařízení posílá autentizační rámec
4. Přístupový bod odpovídá dalším autentizačním rámcem
5. Zařízení zasílá asociační požadavek
6. V případě, že přístupový bod souhlasí, odpovídá zprávou potvrzující asociaci
7. Následně může začít přenos dat

Po těchto krocích následují další autentizační kroky s hesly, které jsou na obrázcích 2.1 a 2.2.

2.1 WEP

Wired Equivalent Privacy (WEP) [1] byl schválen jako protokol Wi-Fi zabezpečení v roce 1999 [10]. Původně měl poskytovat ekvivalentní zabezpečení jako kabelové připojení, nicméně se později zjistilo, že má velké bezpečnostní chyby.

I přes to, že byl v roce 2003 vyvinut nový bezpečnější protokol (WPA), a v roce 2004 od WEP oficiálně Wi-Fi Aliance ustoupila, můžeme dodnes nalézt sítě, které WEP využívají, viz statistika[1].

[1] https://wigle.net/stats#

Šifrování

WEP používá proudovou RC4 šifru pro zajištění soukromí a kontrolní součet CRC-32 na zajištění integrity zprávy [14]. Nejdříve je nutná výměna klíče k mezi uživateli a sítí (WEP protokol nespecifikuje výměnu). Klíč má 40 nebo 104 bitů. V případě 40bitového klíče se jedná o 10 hexadecimálních čísel (5 ASCII znaků), a u delšího klíče pak o 26 hexadecimálních čísel (13 ASCII znaků).

Pro poslání zprávy je nutné vypočítat kontrolní součet zprávy a připojit ho ke zprávě [9]. Výsledná hodnota je zašifrována pomocí RC4, kde se využije klíč k a veřejný inicializační vektor (IV) o délce 24 bitů. Výsledek této operace je poslán po síti, a příjemce, který zná klíč k, ho může dešifrovat.

- **RC4**

  Proudová šifra RC4 je založena na dvou algoritmech: `Key Scheduled Algorithm` (KSA) a `Pseudo Random Generation Algorithm` (PRGA) [9].

  KSA slouží k převedení klíče (1 až 256 bitů) na počáteční permutaci S čísel od *0* po *N*. Čísla i a j určují vnitřní stav RC4 a slouží jako ukazatele na prvky S.

Algorithm 1 KSA – převedení klíče na počáteční permutaci

```
for i = 0, ..., N − 1 do                                   ▷ Inicializace
    S[i] = i
end for
j = 0
for i = 0, ..., N − 1 do                                   ▷ Permutace
    j = j + S[i] + K[i mod l]      ▷ l - počet slov klíče K, kde každé slovo má n bitů
    Swap(S[i], S[j])
end for
```

Algorithm 2 PRGA – generování keystreamu

```
i = 0                                                      ▷ Inicializace
j = 0                                                      ▷ Generační cyklus
while true do
    i = i + 1
    j = j + S[i]
    Swap(S[i], S[j])
    Output z = S[S[i] + S[j]]                              ▷ Výstup
end while
```

  PRGA podle aktuálního vnitřního stavu vygeneruje jeden bajt keystreamu a následně aktualizuje vnitřní stav. Této vlastnosti je využito při útocích na sítě se zabezpečením WEP, viz 3.1.

- **CRC-32**

  Pro zajištění integrity zprávy se v protokolu WEP využívá hešovací funkce z rodiny Cyclic Redundancy Check (CRC) [9].

  Originální zpráva se XORuje konstantou o délce 32 bitů doplněnou sekvencí nul, aby měla stejnou délku jako zpráva. Výsledek pak tvoří novou zprávu a proces se

opakuje, dokud je délka výsledku větší než délka konstanty. Na závěr se tento kontrolní součet připojí ke zprávě. Provedení kontroly se provádí stejným způsobem, avšak s tím rozdílem, že vstup je celá zpráva i s CRC a výsledek není kontrolní součet ale samé nuly, pokud byla integrita zachována.

Autentizace

Pro autentizaci při WEP zabezpečení existují dvě metody: Shared key authentication a Open System authentication [14].

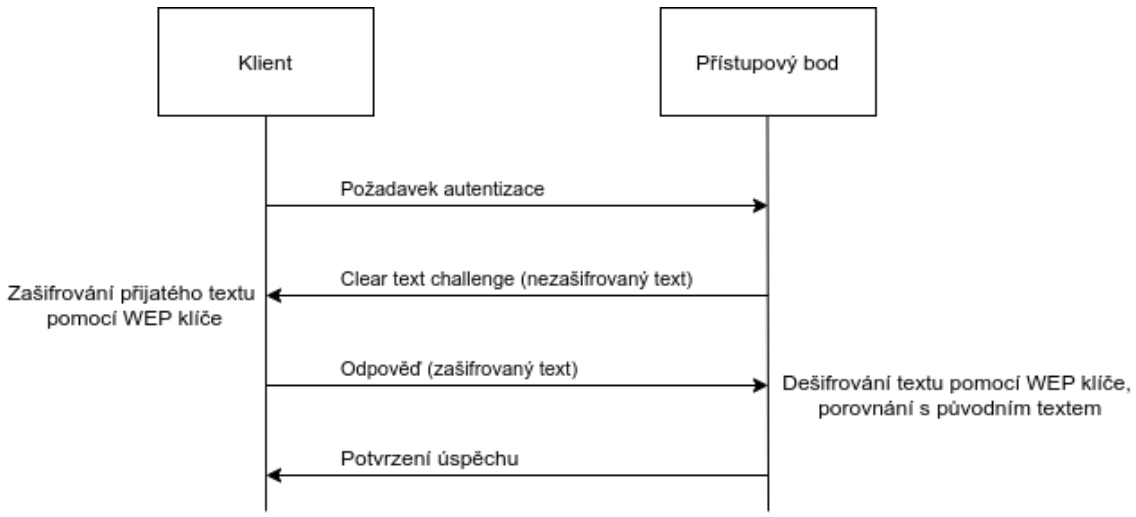
Obecně se jako bezpečnější způsob považuje Open System, jelikož zde klient neposkytuje přístupovému bodu přihlašovací údaje. Díky tomu se klíč může používat na zašifrování posílaných paketů. Stačí tedy, aby klient měl správný klíč.

Autentizace u Shared key sestává ze 4 kroků (obrázek 2.1):

1. Klient pošle autentizační požadavek.
2. Přístupový bod pošle clear text challenge.
3. Klient zašifruje clear text a pošle jej zpátky.
4. V případě, že obsah je po dešifrování stejný, přístupový bod odešle pozitivní odpověď, a klient je tím asociován s přístupovým bodem (AP).

Clear text challenge spočívá v tom, že přístupový bod pošle nezašifrovaný text, poté jej klient zašifruje svým klíčem a pošle zpátky. Zde se po dešifrování pozná, jestli je obsah textu stejný. Pokud ano, byl použitý stejný klíč.

V tomto případě se klíč využije pro autentizaci, a jelikož je při autentizaci nutná výměna paketů, je možné zachytit tuto komunikaci a následně z ní klíč získat.



Obrázek 2.1: Autentizace WEP (Shared key) [21]

2.2 WPA

Po nasazení protokolu WEP se objevily jeho výrazné nedostatky pramenící z použití RC4 a krátkého hesla, bylo nutné vymyslet nový, a tak se zrodil Wi-Fi Protected Access (WPA) [5].

WPA byl však implementován tak, aby jej mohly používat zařízení se stávajícím hardware [18]. Proto jsou přenášené pakety opět šifrované pomocí šifry RC4 [6]. Nicméně na rozdíl od WEP zde byl doimplementován protokol TKIP (Temporal Key Integrity protocol), který se snaží napravit bezpečnostní chyby protokolu WEP, např. opakování inicializačních vektorů [9]. Avšak kvůli použití stejného šifrování (RC4) je stále zranitelný.

TKIP - vylepšení protokolu WEP [9]

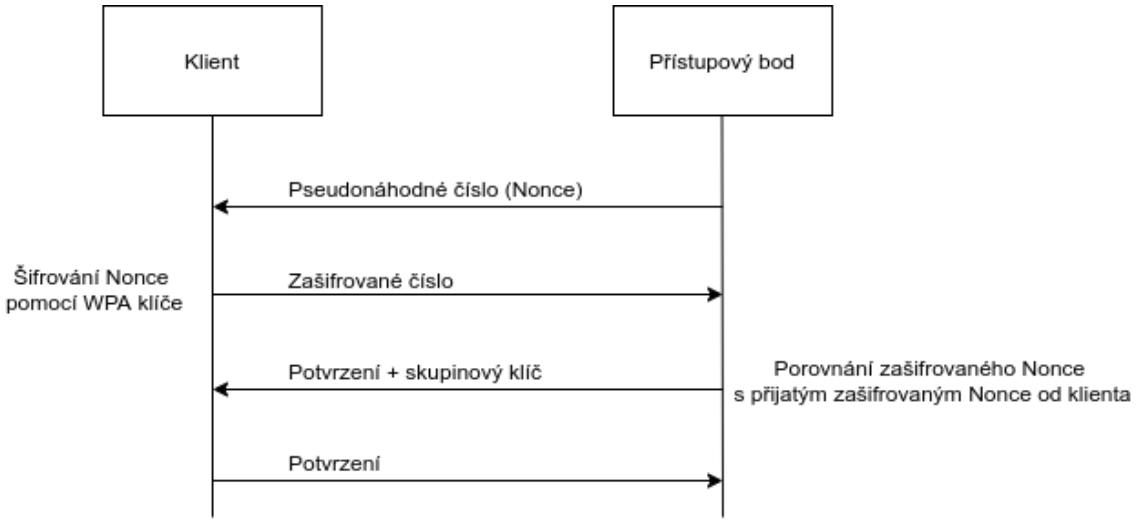
- Message Integrity Check (MIC) zde nahrazuje kontrolní součet CRC-32 používaný u WEP. MIC využívá algoritmus Michael, který hešuje každý paket jeho obsahem a klíčem. Útočník tak bez znalosti klíče nemůže měnit ani falšovat pakety.
- WPA je doplněn o opatření proti brute-force útokům. V případě, že přístupový bod obdrží dvě zprávy se špatným MIC (heš zprávy nesedí s jejím obsahem) za jednu minutu, odpojí všechna připojená zařízení.
- TKIP Sequence Counter nahrazuje starší inicializační vektory. Jedná se o 48bitové číslo, které se zvyšuje pro každý paket. Poskytuje tak ochranu proti opakovanému používání inicializačních vektorů a chrání tak před zjištěním hesla, viz 3.1.
- Na rozdíl od WEP je pro každý paket vytvořen nový klíč pro šifrování, nejedná se už o pouhé spojení klíče a inicializačního vektoru.

Autentizace

Typy WPA autentizace se dělí do dvou tříd, WPA-Personal a WPA-Enterprise.

- **WPA-Personal**

  WPA-Personal, nebo také WPA-PSK (Pre-Shared Key) je navržený pro užití v domácnostech nebo menších firmách [16]. Pro autentizaci se zde využívá sdílený klíč, který musí znát klient i přístupový bod. Všichni uživatelé používají stejný přístupový klíč. Autentizace probíhá v následujících krocích a nazývá se 4-way handshake [17].



Obrázek 2.2: Autentizace WPA-Personal [17]

1. Přístupový bod zašle klientovi pseudonáhodné číslo (typicky označované jako Nonce).
2. Klient toto číslo zašifruje pomocí WPA klíče a pošle zpátky přístupovému bodu.
3. Přístupový bod sám zašifruje číslo WPA klíčem a výsledek porovná s tím, který přijal od klienta. V případě, že se rovnají, má uživatel správný WPA klíč.
4. Klient potvrzuje výměnu.

- **WPA-Enterprise**

  Tento způsob autentizace je navržen pro velké sítě např. v podnicích nebo školách [16]. Na rozdíl od WPA-Personal vyžaduje autentizační RADIUS server. Ověření přes RADIUS je založeno na certifikátech a neprobíhá přímo na routeru. Toto ověření je bezpečnější, ale celkově RADIUS vyžaduje komplikovanější nastavení, proto se pro běžné domácí použití nehodí. Při použití tohoto ověření dostane každý uživatel svoje přihlašovací údaje, pomocí kterých se následně připojuje.

  Útoky na RADIUS jsou samostatnou kapitolou, mnohem složitější než útoky na Wi-Fi sítě a nad rámec této práce. Prakticky zde není možné certifikát prolomit pomocí brute-force a je nutné používat útoky typu Man in the Middle.

Pro šifrování se používají klíče o délce 256 bitů [10]. Díky použití delšího klíče je u stejné šifry jak u WEP dosaženo lepších výsledků.

2.3 WPA2/WPA3

WPA byl v roce 2006 oficiálně nahrazen novým protokolem WPA2 [10]. WPA2 [2] přidává k šifrování pomocí TKIP a Michael algoritmu (zahnutém v MIC) nový algoritmus CCMP (Counter Mode with Cipher Block Chaining Message Authentication Code Protocol), nahrazující TKIP, který je ale zachován kvůli zpětné kompatibilitě [6]. Je založen na šifrování AES (Advanced Encryption Standard).

Dnes je už možné setkat se s dalším nástupcem, a to WPA3, který do standardu přináší dvě změny - Simultaneous Authentication of Equals (SAE) a Management Frame Protection (MFP) [4], [3]. WPA3 bylo představeno v roce 2018, a od 1. července 2020 musí všechna nová Wi-Fi zařízení splňovat tento standard [13]. WPA3 přináší nové, robustnější šifrování pomocí AES s GCMP (Galois/Counter Mode Protocol). Velkou výhodou WPA3 je přidání MFP, díky kterému je možné bránit se např. proti deautentizačním útokům [15], viz kapitola 5.

Pre-Shared Key je zde nahrazen SAE, který je založen na Dragonfly handshake [13]. Tím WPA3 eliminuje problémy, které s PSK měly předchozí verze WPA.

WPS

Ve standardu WPA byl implementován Wi-Fi Protected Setup (WPS) [19]. Jedná se funkci přístupového bodu, implementovanou na WPA/WPA2-PSK pro zjednodušení připojování se k Wi-Fi. Díky ní je možné místo celého hesla zadat jen osmimístný PIN, nebo během připojování stisknout tlačítko (Push button metoda) na přístupovém bodu. Tato funkce byla zařazena s dobrým úmyslem, pro ulehčení připojování se k síti, implementace však měla řadu slabin, a díky tomu podlamovala bezpečnost WPA, více viz 3.3.

Kapitola 3

Popis prolamování zabezpečení Wi-Fi sítí

Útoků na Wi-Fi existuje celá řada, tato práce se zaměřuje především na útoky, pomocí kterých je možné získat přístupové heslo k Wi-Fi síti. U zabezpečení WEP se může jednat o pouhý brute-force útok, existují ale také další více sofistikované útoky, které využívají zejména slabiny šifrování.

Pro novější protokoly je použití brute-force útoků komplikované výpočetní náročností, proto jsou používané útoky vedeny zejména na konkrétní části protokolu zabezpečení. U WPA se jedná například o PMKID útok, dalšími z této kategorie jsou například útoky na funkci WPS. Hojně používaným útokem v dnešní době je slovníkový útok na zabezpečení WPA/WPA2, jelikož většina útoků, které cílí na slabiny protokolu, již byla opravena.

3.1 Útoky na zabezpečení WEP

WEP byl prvním zabezpečením Wi-Fi sítí a dle očekávání se v něm nachází nejvíce slabin, je tak nejjednodušeji prolomitelný. Tim Newsham, výzkumník v oblasti bezpečnosti, nalezl u některých výrobců chybu, která umožňuje použití brute-force metody k prolomení hesla [7].

Tato slabina se nachází v generátoru klíče u některých routerů při použití 40bitového klíče, viz 2.1. Uživatel má při zadávání hesla možnost místo hexadecimálních čísel zadat klíč jako text ve formátu ASCII, a následně jsou znaky z textu interpretovány jako hexadecimální čísla. Chyba spočívá v tom, že při tomto převodu se použije generátor pseudonáhodných čísel se 32bitovou počáteční hodnotou. Jelikož ale nejvyšší bit ASCII hodnoty je vždy 0 a generátor používá operaci XOR na ASCII hodnoty, zjistilo se, že z původních 32bitových počátečních hodnot můžeme nyní počítat pouze s 21bitovými hodnotami. A tak tedy brute-force metoda použitá na počítači, který zvládne 60000 odhadů hesla za sekundu, bude trvat zhruba 35 sekund.

Automatické generování hesel tedy není doporučováno, uživatel by měl 10 hexadecimálních čísel vyplnit sám, a tím mnohonásobně sníží pravděpodobnost prolomení touto metodou (zde se dostáváme na čas zhruba 210 dní, ale čas v řádu stovek dní není dostatečný, jelikož úloha je lehce paralelizovatelná a i bez použití cloud computingu a superpočítačů je možné do týdne prolomit jakékoli heslo na výkonnějším serveru. V případě použití klíče o délce 104 bitů už se tato chyba u generování neobjevuje a odhadovaná délka pro použití brute-force je 10^{19} let (s již zmíněným počítačem, při použití moderních počítačů by tato

doba byla mnohem kratší). Nicméně brute-force útok na WEP není jediný. Další známé útoky, fungující na jiném principu, jsou popsány níže.

FMS

Útok FMS je pojmenovaný podle trojice Fluhrer, Mantin a Shamir, která jej publikovala v roce 2001 [20]. Tento útok využívá slabiny šifrování RC4 k získání hesla. Jelikož první bajty paketu jsou lehce předpověditelné, je možné získat první bajty keystreamu, pomocí kterého je paket zašifrován. Mimo jiné se inicializační vektory posílají nezašifrované, a tak útočník, který pakety zachytává, může ihned zjistit tři bajty klíče pro každý paket [9]. V případě, že platí určité podmínky, je útočník schopný odhadnout s 5% šancí jeden bajt klíče. Následně je možné na základě daného bajtu dopočítat následující, a tak dále. Je tedy možné takto odhadnout celý klíč, vyzkoušet ho, a v případě neúspěchu proces opakovat (útok je detailněji popsán níže). Aby byl útok splněný s 50% úspěšností, je nutné nasbírat zhruba 6 000 000 paketů.

Metoda popsaná v článku počítá s tím, že útočník zná prvních l bajtů klíče pro paket. Je pak snadné provést prvních l kroků KSA algoritmu, viz 2.1. Útočník tedy zná S_l a j_l, následně $j_{l+1} = j_l + K[l] + S_l[l]$ a S_l se prohodí s $S_l[j_{l+1}]$.
Pokud budou splněny následující podmínky:

- $S_l[1] < l$
- $S_l[1] + S_l[S_l[l]] = l$
- $S_l^{-1}[X[0]] \neq 1$
- $S_l^{-1}[X[0]] \neq S_l[1]$

Pak $S_{l+1}[l]$ nabude v dalším kole KSA hodnotu $S_l[j_{l+1}]$ a hodnota $S[l]$ se během zbytku procesu nezmění s pravděpodobností zhruba 5 %. Případně tedy první bajt keystreamu $X[0]$ bude $S[l]$, a díky tomu bude mít útočník možnost dopočítat následující bajt klíče K:

$$K = S_l^{-1}[X[0]] - j_l - S_l[l] = S_l^{-1}[S_{l+1}[l]] - j_l - S_l[l]$$

Jelikož pravděpodobnost úspěchu je pouze 5 %, provede útočník tento výpočet na několika paketech a vybere nejpravděpodobnější K jako další bajt klíče. Poté tuto činnost provádí inkrementálně, a dostává se tak v každém dalším kroku o bajt dál. Následně klíč vyzkouší, a pokud nebude správný, pouze vymění zvolený K za jiný hodně pravděpodobný a proces opakuje.

KoreK

Útok KoreK, publikován v roce 2004, je pojmenován po přezdívce jeho autora [9]. Využívá 16 dalších korelací mezi prvními l bajty RC4 klíče, prvními dvěma bajty vygenerovaného keystreamu a následujícím bajtem klíče. Nějaké z těchto korelací byly známy veřejnosti již dříve, nicméně většinu z nich nalezl KoreK. Téměř většina z těchto korelací využívá vlastnost, že první nebo druhý bajt keystreamu odhaluje j_{l+1} za daných podmínek.

Tento útok je velmi podobný FMS útoku, na rozdíl od něj je však mnohem efektivnější. Pro zajištění 50% pravděpodobnosti úspěchu zde stačí pouze 700 000 zachycených paketů.

PTW

Útok PTW pojmenovaný podle Pyshkina, Tewse a Weinmanna byl publikovaný v roce 2007 a přináší dva nové koncepty [9]:

1. V roce 2005 přišel Klein na to, že $l - X[l-1]$ nabude hodnoty $S[l]$ s pravděpodobností 2/256. V případě, že se $S[l]$ nezmění do doby, než se vytvoří $X[l-1]$, pak platí, že:

$$S_l^{-1}[l - X[l-1]] - (S_l[l] + j_l) \tag{3.1}$$

   nabude hodnoty $K[l]$ s pravděpodobností 2/256. V opačném případě bude mít výraz 3.1 náhodnou hodnotu. Můžeme tedy říct, že výraz 3.1 bude mít hodnotu $K[l]$ s pravděpodobností přibližně 1.37/256.

   Tato vlastnost může být na rozdíl od metod FSM nebo KoreK použita na jakýkoli paket.

2. PTW útok pracuje s vícebajtovou korelací namísto hádání klíče bajt po bajtu. Za předpokladu, že útočník zná prvních l bajtů klíče a dopočítá si $k = S_{l+2}[l+1]$, může pak použít vlastnost $S_{l+1}^{-1}[k] - S_{l+1}[l+1] - S_l[l] - j_l = K[l] + K[l+1]$.

   Phiskin, Tewes a Weinmann tohoto vztahu využili a upravili výraz 3.1 tak, aby nevyjadřoval jeden bajt klíče, ale součet m následujících bajtů klíče kde $m \in \langle 1, 13 \rangle$. Píšeme $\sigma_i = \sum_{k=0}^{i} Rk[k]$, a platí vztah:

$$S_l^{-1}[l + m - 1 - X[l + m - 2]] - \sum_{a=l}^{l+m-1} S_l[a] \tag{3.2}$$

   závisející pouze na inicializačních vektorech, který dává váhy všem σ_i.

Toho se při útoku využije následovně [20]:

1. Nejdříve útočník zachycuje pakety a získává z nich keystream (obdobně jak u FMS), z toho plyne, že útočník zná první 3 bajty všech klíčů pro paket.

2. Poté útočník vypočítá hodnotu výrazu 3.2 pro všechny pakety a všechna m, a získá ohodnocení pro každé σ_i.

3. Po zpracování všech paketů se klíč získává následovně:

$$Rk[0] = \sigma_0 \quad a \quad Rk[i] = \sigma_i - \sigma_{i-1}$$

4. V případě, že je vypočítané heslo nesprávné, stačí vyměnit hodnotu σ_i za jinou pravděpodobnou a vypočítat znovu.

Pro dosáhnutí 50% úspěšnosti stačí zachytit 35 000 až 40 000 paketů [20].

- Útoky PTW i KoreK je možné spustit v nástroji Aircrack-ng 4.1 pomocí přepínačů `-z` a `-K`.

3.2 Útoky na zabezpečení z rodiny WPA

Níže jsou popsány útoky na zabezpečení WPA popsané v kapitole 2.2. Nejznámějším a také v dnešní době hojně používaným útokem je Slovníkový útok, proti kterému je jediná ochrana dostatečně unikátní a dlouhé heslo. Dalším z útoků je PMKID útok, který je ale dnes na téměř všech přístupových bodech opraven. Známým útokem je také KRACK, který cílí na znovupoužití stejného klíče k šifrování zprávy. Pomocí toho je možné následně dešifrovat komunikaci, ale nelze jím zjistit heslo k přístupovému bodu.

Slovníkový útok

Aby mohl útočník provést slovníkový útok, je nutné, aby nejdříve zachytil komunikaci mezi připojujícím se zařízením a přístupovým bodem [12]. Během této komunikace si zařízení s přístupovým bodem vymění zahešovaný klíč viz 2.2. V případě pasivního útoku bude útočník jen zachytávat provoz a čekat, až se někdo k dané síti připojí, aby mohl zachytit 4-way handshake. To ale může trvat delší dobu, a tak je možnost provedení aktivního útoku (Deauthentication attack), pomocí kterého donutí útočník zařízení se odpojit. Poté, co se zařízení opět připojí, může útočník zachytit 4-way handshake. Deautentizační útok může být zaznamenaný i samotným uživatelem, jelikož je ze sítě odpojen. Nicméně se nejedná o závažný problém, jelikož k podobnému odpojení zařízení od sítě může dojít i v běžném provozu, a ve většině případů se zařízeni velmi rychle připojí znovu. Možnosti obrany proti tomuto útoku jsou popsány v kapitole 5.

Po zachycení handshake následuje slovníkový útok pro získání hesla. Během této části už není nutné být v dosahu sítě.

Slovník pro účely crackování je textový soubor obsahující hesla nebo jejich fragmenty, které budou při prolamování použity. Pro účely crackování se většinou využívají slovníky, které obsahují uniklá hesla. Aby byl útok úspěšný, je nutné, aby konkrétní heslo, které se snažíme prolomit, bylo obsaženo v daném slovníku, který pro útok používáme. Proto je pro tento typ útoku velmi důležité mít správný slovník (např. pokud použijeme slovník obsahující česká slova a budeme se pokoušet prolomit síť, která bude mít pravděpodobně heslo anglické, je velmi pravděpodobné, že dané heslo neprolomíme).

Slovníky je možné je najít v různých distribucích určených pro penetrační testování (např. v *Kali Linux* se slovníky nacházejí ve složce `/usr/share/wordlists/`[1]).

PMKID útok

Hlavní výhodou tohoto útoku je, že na síti nemusí být připojen žádný uživatel [11]. Útočník naruší proces autentizace se sítí, viz kapitola 2 tak, že zašle síti asociační požadavek v momentě kdy ještě není autentizován. Na tuto žádost některé routery odešlou paket, který obsahuje PMKID (Pairwise Master Key Identifier). V případě, že se útočník dostane k PMKID, může ho následně prolomit pomocí slovníkového útoku obdobně jako se zachyceným handshake. Tento útok je možné provést pomocí nástrojů Airgeddon a Wifite, popsaných v kapitole 4. Tento útok byl však už na většině routerů, podobně jako útoky na WPS, opraven.

[1] https://tools.kali.org/password-attacks/wordlists

3.3 Útoky na WPS

U zařízení, které využívají WPS je povinný osmimístný PIN, a jeho kontrola probíhá po čtyřech číslech [19]. Přístupový bod tedy nejdříve zkontroluje první čtyři čísla PINu a následně další tři (poslední číslo slouží jako kontrolní součet). Z čehož vychází celkem 11 000 kombinací ($10^4 + 10^3$). Některé přístupové body umožňují testovat přístupový PIN i po nesprávné kombinaci bez časového odstupu, je zde tedy lehké použít brute-force metodu. Další možností je použití Pixie Dust útoku.

Pixie Dust

Dalším útokem na WPS je útok objevený Dominiquem Bongardem, který využívá slabinu některých routerů při generování náhodných čísel. Jedná se například o routery Broadcom a Ralink [8]. Databázi všech routerů, které mají tuto slabinu, je možné najít na adrese[2].

Pro Pixie Dust útok jsou velmi důležité zprávy *M1* a *M3* [8], viz obrázek 3.1.

Ve zprávě *M1* posílá přístupový bod klientovi Nonce hodnotu a veřejnou část klíče `PKE`. Ve zprávě *M3* posílá přístupový bod dvě hešované hodnoty (`E-Hash1` a `E-Hash2`), které jsou vygenerovány následovně:

- `E-Hash1` = HMAC-SHA-256(authkey)(E-S1 | PSK1 | PKE | PKR)
- `E-Hash2` = HMAC-SHA-256(authkey)(E-S2 | PSK2 | PKE | PKR)

Kde `PKE` je veřejná část klíče přístupového bodu, `PKR` je veřejná část klíče klienta, `PSK1` je tvořen prvními čtyřmi číslicemi PINu a `PSK2` je tvořen posledními čtyřmi čísly PINu.

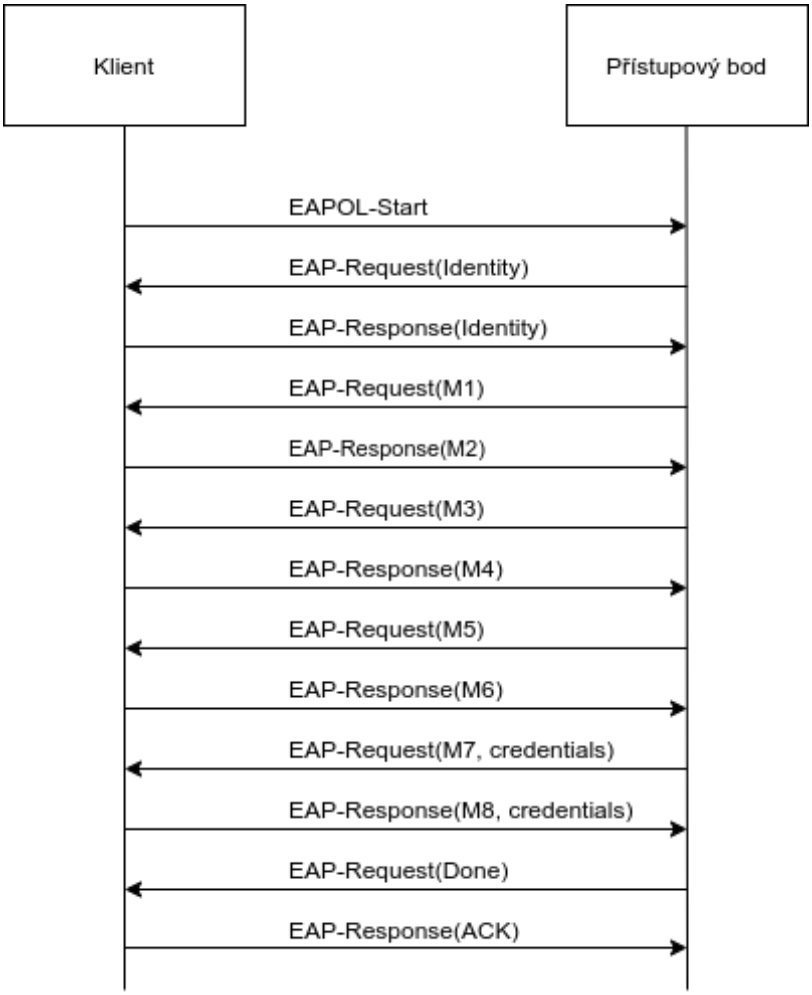
Pokud tedy uživatel zachytí tuto komunikaci, stačí mu následně dopočítat hodnoty `E-S1` a `E-S2`, které jsou pseudonáhodně generovány, aby mohl pomocí brute-force metody zjistit `PSK1` a `PSK2`. Právě zde využívá útok slabiny generování těchto hodnot.

Provedení útoku [8]:

1. Provedení WPS protokolu po zprávu *M3* (včetně)
2. Zjištění hodnot `E-S1` a `E-S2` (díky znalosti implementace generátorů nebo přímo hodnot)
3. Zjištění `PSK1` a `PSK2` pomocí brute-force metod
4. Provedení celého WPS protokolu pro zjištění přihlašovacích údajů

Například zařízení Broadcom vytváří `E-S1` a `E-S2` pomocí generátoru, který generuje Nonce ze zprávy *M1*, proto tedy stačí zjistit jeho stav, a poté tyto hodnoty dopočítat. Dalším z příkladů mohou být zařízení Railink, u kterých jsou hodnoty `E-S1` a `E-S2` vždy nula.

[2] https://docs.google.com/spreadsheets/d/1tSlbqVQ59kGn8hgmwcPTHUECQ3o9YhXR91A_p7Nnj5Y/edit?usp=sharing



Obrázek 3.1: vyjednávací proces WPS [8], Extensible Authentication Protocol (EAP), Extensible Authentication Protocol over LAN (EAPOL)

Kapitola 4

Existující nástroje a jejich výběr

Cílem této práce je vytvoření nástroje, který bude využitelný pro automatizované prolamování Wi-Fi sítí. Jelikož pro většinu útoků, které se používají, byly už nějaké nástroje vytvořeny, bylo nutno zjistit, jaké existují, a jestli by bylo možné je zahrnout do výsledného nástroje.

Proto v této kapitole budou popsány nástroje, které umožňují útoky na Wi-Fi sítě, a bylo by je možné zahrnout do výsledného nástroje. Protože se tato práce primárně zabývá prolamováním hesel Wi-Fi sítí, bude z nástrojů, které pokrývají širší spektrum služeb, popsána pouze tato část.

K penetračnímu testování existují speciální Linuxové distribuce (např. Kali Linux, Parrot OS atd.). Tyto distribuce jsou udržovány rozsáhlou komunitou lidí, která spravuje a vyhledává nástroje k penetračnímu testování a přidává je do těchto distribucí. Díky tomu jsou nástroje aktuální a je tak možné ze seznamů nástrojů vybrat vhodné. Takovým seznamem je například seznam *Kali Linux Tools Listing*[1]. Dalším zdrojem k vyhledávání nástrojů byla fóra věnující se tematice útoků na bezdrátové sítě. Některé nástroje jsou popsány i ve vědecké literatuře.

Nalezené nástroje byly testovány na improvizované domácí síti. Použito bylo 7 různých routerů a jeden mobilní hotspot. V rámci testování byly zkoušeny útoky na WEP zabezpečení, deautentizační útoky, PMIKD útoky a útoky na WPS. Výsledky testování zranitelností routerů je možné nalézt v kapitole 7.5.

4.1 Sada Aircrack-ng

Aircrack-ng se v roce 2006 stal nástupcem předchozí verze Aircrack (ng - next generation). Jedná se o balíček nástrojů, které se zaměřují na různé oblasti Wi-Fi bezpečnosti, kterými jsou:

- **Monitorování sítí** – nástroje pro zachytávání paketů a export do vybraných formátů
- **Útočení na sítě** – nástroje pro deautentizační útoky, vytváření falešných přístupových bodů, injekce paketů atd.
- **Testování** – zjišťování dostupných schopností konkrétních síťových karet
- **Crackování** – nástroje sloužící k prolamování WEP a WPA-PSK hesel

[1] https://tools.kali.org/tools-listing

Všechny nástroje v této sadě jsou konzolové aplikace, a tudíž je jejich použití ve skriptech snadné. Níže jsou popsány vybrané utility a jejich použití, seznam všech je možné najít na adrese[2].

Airmon-ng

Airmon-ng je utilita, která slouží ke zobrazení dostupných bezdrátových rozhraní a přepínání mezi běžným a monitorovacím režimem na konkrétních rozhraních. Síťová karta musí být v monitorovacím režimu, aby mohla zachytávat všechen okolní provoz. Monitorovací režim umožňuje síťové kartě zachytávat všechny dostupné pakety, které jsou zachyceny bez jakéhokoli filtru. Ne každá síťová karta tento režim podporuje.

- Výpis dostupných rozhraní
  `# airmon-ng`
- Zapnutí monitorovacího režimu na rozhraní *wlan0*
  `# airmon-ng start wlan0`

Airodump-ng

Tento nástroj je primárně určen pro zachytávání paketů. V kontextu prolamování Wi-Fi sítí je jeho hlavním využitím zachytávání inicializačních vektorů pro prolomení sítí se zabezpečením WEP, a také rozpoznání zachycení handshake, a tak je možné ho použít v první části útoku na sítě WPA. Nástroj je také možné použít pro zjišťování informací o dostupných sítích, jako například: BSSID, signál přístupového bodu, počet zachycených paketů z daného přístupového bodu, číslo kanálu, typ zabezpečení a šifrování a mnoho dalších.

- Zobrazení informací o dostupných sítích pomocí rozhraní *mon0*
  `# airodump-ng mon0`
- Zachytávání paketů z konkrétního přístupového bodu (parametr `--bssid`) na konkrétním kanálu (parametr `-c`). Pakety jsou ukládány do souboru *WPAcrack* (paramter --write) pomocí rozhraní *mon0*.
  `# airodump-ng --bssid 34:2C:C4:AC:ED:8E -c 11 --write WPAcrack mon0`

Aireplay-ng

Pomocí tohoto nástroje je možné provádět injekci paketů (posílání konkrétních paketů na danou síť). Využívá se zejména pro generování provozu na dané síti pomocí zasílání speciálních paketů a odpojování uživatelů. Pomocí tohoto nástroje je možné provádět útoky, jako např.: Deautentizace, Falešná autentizace, Fragmentační útok, a další.

Deautentizační útok je velmi důležitý pro prolamování WPA hesel. Pro provedení slovníkového útoku je totiž nutné mít zachycený handshake. Pomocí deautentizace je možno odpojit uživatele od sítě a při jejich opětovném připojení jej zachytit. V případě, že se nepoužije tento útok, musí útočník čekat do doby, než se na síť někdo připojí sám.

Tato část útoku je detekovatelná i samotným uživatelem, neboť uživatel pozná, že byl odpojen od Wi-Fi. Tento jev může být pro uživatele obtěžující, ale zároveň k němu může docházet z různých příčin. Opětovné připojení může také proběhnout velmi rychle, takže jej uživatel ani nemusí zaznamenat.

[2] http://aircrack-ng.org/doku.php

- Deautentizační útok na konkrétní přístupový bod (parametr `-a`), s daným počtem deautentizačních paketů (parametr `--deauth`) pomocí rozhraní *mon0*.
  `# aireplay-ng --deauth 100 -a C8:3A:35:48:D7:E0 mon0`

Aircrack-ng

Nástroj, podle kterého se jmenuje celá sada, slouží pro prolamování WEP a WPA/WPA2-PSK hesel.

Pro crackování WEP hesel umožňuje Aircrack-ng použití dvou metod. První a výchozí metodou je PTW, která je popsána výše, viz 3.1. Druhou metodou je použití starších metod FMS/KoreK, které jsou popsány v kapitole 3.1, přičemž jsou zde metody zkombinovány a doplněny o prostou brute-force metodu.

Při crackování WPA hesel je možné použít pouze metodu slovníkového útoku nad souborem se zachyceným handshake.

- Prolamování WEP hesla ze souboru *WEP.cap*
  `$ aircrack-ng WEP.cap`
- Prolamování WPA hesla ze souboru *WPA.cap*, cesta ke slovníku je zadána parametrem `-w`
  `$ aircrack-ng WPA.cap -w /passwords/dicitonaries/dictionary`

Besside-ng

Tento nástroj slouží k automatizovanému crackování všech WEP sítí v dosahu a zachytávání WPA handshake. Je možné zadat specifikace pro výběr sítí. Při zachycení handshake musí uživatel prolamování provést manuálně, samotný nástroj toto neautomatizuje.

- Crackování všech dostupných WEP sítí a sbírání WPA handshake, pomocí rozhraní *mon0*
  `# besside-ng mon0`
- Zachytávání WPA handshake (parametr `-W`) konkrétního přístupového bodu (parametr `-b`) pomocí rozhraní *mon0*
  `# besside-ng -W -b 00:00:11:22:33:44 mon0`

Ukázka použití - útok na WPA2-PSK

1. Zjištění dostupných síťových rozhraní.
   `# airmon-ng`.

   Výstup:

   ```
   PHY     Interface  Driver   Chipset
   phy0    wlan0      iwlwifi  Intel Corporation Wireless 3160 (rev 83)
   ```

   Zde je pro nás důležité zjistit, jak se jmenuje rozhraní, které chceme použít. Jeho název nalezneme ve sloupci *Interface*.

2. Přepnutí rozhraní *w0* do monitorovacího režimu.
   `# airmon-ng start w0`

   Výstup:

```
Found 2 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode
    PID Name
    584 NetworkManager
    736 wpa_supplicant
PHY     Interface Driver Chipset
phy0    w0       iwlwifi Intel Corporation Wireless 3160(rev 83)
        (mac80211 monitor mode vif enabled for [phy0]w0 on [phy0]mon0)
        (mac80211 station mode vif disabled for [phy0]w0)
```

Zde se z výstupu dozvídáme, že existují dva procesy, které by mohly způsobovat potíže, je možné je zastavit pomocí navrhovaného příkazu, není to však nutné.

Při zapnutí monitorovacího režimu se mění jméno síťového rozhraní. Jak se jmenuje po zapnutí je možné zjistit z předposledního řádku, v tomto případě to je *mon0*.

3. Zjištění informací o dostupných sítích.
   `# airodump-ng mon0`

   Výstup:

```
CH 11 ][ Elapsed: 0 s~][ 2021-02-07 19:04

 BSSID PWR Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID

 XX:XX:XX:12:08:6F -74 4 0 0 11 130 WPA2 CCMP PSK black
 XX:XX:XX:E6:42:68 -48 4 0 0 11 130 WPA2 CCMP PSK BP-asus-test
 XX:XX:XX:C7:9C:69 -78 2 0 0 11 130 WPA2 CCMP PSK UPC1775384
 XX:XX:XX:5C:E4:2C -65 3 0 0 6 270 WPA2 CCMP PSK Wi-FYzomandias
 XX:XX:XX:7A:B7:0D -74 3 0 0 4 270 WPA2 CCMP PSK dlink-B70C

 BSSID STATION PWR Rate Lost Frames Notes Prob

 XX:XX:XX:AC:ED:8E XX:XX:XX:75:43:76 -89 0 - 1 0 1
 (not associated) XX:XX:XX:13:DF:FC -81 0 - 1 11 3
```

   Z tohoto výstupu jsou pro nás důležité dvě informace. Jedná se o MAC adresu přístupového bodu, na který chceme útočit, najdeme jej ve sloupci *BSSID*, dále je pro nás důležité číslo kanálu, to nalezneme ve sloupci *CH*. Tyto informace použijeme v dalším kroku. Jakmile zjistíme potřebné informace o cílovém přístupovém bodu, můžeme vykonávání programu zastavit.

4. Zaměření se na konkrétní přístupový bod, v našem případě *BP-asus-test* s konkrétním kanálem (*11*), zachycená komunikace se bude ukládat do souboru *WPA*.

   `# airodump-ng --bssid 10:BF:48:E6:42:68 -c 11 --write WPA mon0`

   Výstup:

```
CH 1 ][ Elapsed: 24 s~][ 2021-02-07 19:16 ][ WPA handshake: MAC
 BSSID PWR RXQ Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID
```

```
10:BF:48:E6:42:68 -46 100 263 9 2 1 130 WPA2 CCMP PSK BP-asus-test

BSSID STATION PWR Rate Lost Frames Notes Probes

10:BF:48... C2:4C:10:C6:AF:42 -28 1e- 1 101 92 PMKID BP-asus-test
```

V případě, že provádíme útok na WPA-PSK, je pro nás důležitá pouze jedna věc, a to zachycení handshake. To poznáme z prvního řádku, dokud se na konci neobjeví *WPA handshake: MAC* (konkrétní MAC adresa), nebyl handshake zachycen.

5. Zasílání deautentizačních paketů na daný přístupový bod slouží k odpojení připojených zařízení, aby bylo snazší zachytit handshake, až se zařízení budou opět připojovat. Tento krok je volitelný, v případě, že nám nezáleží na tom, jak dlouho bude trvat zachytávání handshake, můžeme jej přeskočit, může to ale také trvat několik dní (dokud se někdo na síť sám nepřipojí, ani to však negarantuje zachycení handshake). Tento krok provádíme zároveň s předchozím krokem.
   ```
   # aireplay-ng --deauth 100 -a 10:BF:48:E6:42:68 mon0
   ```
   Výstup:
   ```
   19:32:01 Waiting for beacon frame (BSSID: MAC) on channel 11
   NB: this attack is more effective when targeting
   a connected wireless client (-c <client's mac>).
   19:32:01 Sending DeAuth (code 7) to broadcast -- BSSID: [MAC]
   19:32:02 Sending DeAuth (code 7) to broadcast -- BSSID: [MAC]
   19:32:02 Sending DeAuth (code 7) to broadcast -- BSSID: [MAC]
   ```

6. V případě, že byl zachycen handshake, můžeme přistoupit na prolomení hesla pomocí slovníkového útoku. Tuto část útkou už lze provádět offline.
   ```
   $ aircrack-ng WPA-01.cap -w ./skola/BP/slovniky/rockyou.txt
   ```
   Výstup:
   ```
                              Aircrack-ng 1.6

         [00:00:01] 5374/10303727 keys tested (8931.47 k/s)

         Time left: 19 minutes, 13 seconds                          0.05%

                              KEY FOUND! [ test1234 ]


         Master Key     : 1E 46 72 EB C5 0C 82 C7 2F 92 D6 40 7D CB AB CD
                          C8 C6 1A 0F 89 1E BD D2 14 EC 55 A7 40 50 D9 0D

         Transient Key  : 81 95 4D 80 F7 1C B5 59 07 92 D9 B3 A6 86 F4 C3
                          75 E4 3D A9 26 48 C0 52 81 BA 35 B7 71 C2 27 6E
                          77 66 22 C4 7C 5E B9 67 9B A2 B3 30 58 80 A8 00
                          00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
   ```

```
EAPOL HMAC     : FE EE AA 86 D3 CD B7 E3 C9 DD 55 90 5A 30 AF 70
```

Pokud bylo hledané heslo v našem slovníku, najdeme ho ve výpisu v hranatých závorkách. Tento krok může být časově velmi náročný a nemusí doběhnout úspěšně, pokud heslo ve slovníku vůbec nebylo.

4.2 LAZY script

LAZY script obsahuje mnoho nástrojů sloužících k penetračnímu testování, jejichž kompletní seznam je možné nalézt na adrese[3]. LAZY script zabaluje různé nástroje a poskytuje jim unifikované API v příkazové řádce. Tím automatizuje a zjednodušuje používání těchto nástrojů. Jelikož jsou zde útoky předdefinovány, stačí zvolit útok a vybrat přístupový bod, následně bude útok proveden.

Po spuštění uvítá uživatele obrazovka s dostupnými možnostmi. Kromě útoků pro získání hesla přístupového bodu zde uživatel nalezne i další funkce (viz ukázka níže), a mimo jiné i přepnutí rozhraní do monitorovacího módu. Uživatel vybírá varianty pomocí příslušných písmen nebo čísel, po zvolení se pak zobrazí nové okno terminálu s další nabídkou. Pomocí možnosti *TOOLS* je možné doinstalovat další penetrační nástroje.

Pro účely této práce je ze skriptu možné využít nástroje pro zachycení handshake a následné prolomení hesla pomocí slovníkového útoku, nebo nástroje pro prolomení WEP zabezpečení. U obou těchto útoků využívá skript nástroje ze sady Aircrack-ng, viz 4.1.

- Ukázka dostupných možností:

if) Ifconfig	l) Local IPs & gateways	scan) Arp-scan network
1) Enable wlan0	d1) Disable wlan0	start) Start monitor mode
2) Enable wlan0mon	d2) Disable wlan0mon	stop) Stop monitor mode
3) Change MAC	d3) Restore original MAC	update) Check for updates
4) Enable anonym8	d4) Disable anonym8	errors) Fix some errors
5) Enable anonsurf	d5) Disable anonsurf	ks) Keyboard shortucts
6) Anonsurf's status	d6) Restart anonsurf	d) Buy me a coffee
7) View public IP		s) Go to settings menu
8) View MAC		
9) TOOLS	15) Spoof EMAIL	22) Show bandwidth
10) Handshake	16) Ngrok port forward	
11) Find WPS pin	17) Ask (Howdoi tool)	
12) WEP menu	18) Auto-exploit browser	
13) MITM	19) Geolocate an IP	
14) Metasploit	20) Bruteforce login	
0) Exit	21) Sqlmap automated	

4.3 Airgeddon

Dalším z automatizačních nástrojů je Airgeddon. Jeho použití je velmi podobné výše popsanému LAZY scriptu. Uživatel zde pomocí čísel volí možnosti daného útoku, a podle

[3] https://github.com/arismelachroinos/lscript

jeho volby se mu následně ukazují další možnosti. Kromě nástrojů, sloužících pro získání hesla přístupového bodu, zde nalezneme i nástroje pro DoS útoky, nebo nástroje pro Evil Twin útoky (viz ukázka níže). Bohužel na rozdíl od LAZY scriptu nemá Airgeddon možnost crackování hesel. Slouží tedy pouze k nachytání následně použitelného materiálu pro získání hesla. Airgeddon při spuštění automaticky doinstaluje veškeré nástroje, které je možné v něm použít, a tak se uživateli mohou nainstalovat i nástroje, které nikdy nepoužije. Toto může způsobovat problémy v zařízeních, které nemají všechny tyto nástroje dostupné.

Pro potřeby této práce skript opět využívá nástroje ze sady Aircrack-ng popsané v 4.1, nicméně je obohacen i o další nástroje, které se zaměřují na jiné bezdrátové útoky. Více informací je možné nalézt na odkaze[4].

- Ukázka dostupných možností:

```
***************************** airgeddon main menu ************************
Interface wlan0 selected. Mode: Managed. Supported bands: 2.4Ghz, 5Ghz
Select an option from menu:
---------
0. Exit script
1. Select another network interface
2. Put interface in monitor mode
3. Put interface in managed mode
---------
4. DoS attacks menu
5. Handshake/PMKID tools menu
6. Offline WPA/WPA2 decrypt menu
7. Evil Twin attacks menu
8. WPS attacks menu
9. WEP attacks menu
10. Enterprise attacks menu
---------
11. About & Credits
12. Options and language menu
---------
```

4.4 Wifite2

Nástroj Wifite2 je nástupce a vylepšení předchozí verze Wifite. Na rozdíl od předchozích nástrojů je Wifite2 zaměřen na získání hesla přístupového bodu. Podobně jako ostatní nástroje využívá sadu Aircrack-ng, viz 4.1, ale je také obohacen o další nástroje.

V případě, že uživatel nespecifikuje konkrétní útok, spustí se postupně všechny od nejjednodušších po složitější útoky. Uživatel tak tedy může jen nechat zapnutý skript, a po nějaké době zjistí, jestli se podařilo konkrétní přístupový bod prolomit. Wifite2 má několik výhod oproti ostatním nástrojům, například, možnost provedení útoku na více přístupových bodů najednou. Další výhodou je, že skript neotevírá další okna terminálu, což zjednodušuje jeho použití přes *SSH*.

[4] https://github.com/v1s1t0r1sh3r3/airgeddon

Jeho nevýhodou ale je, že při nespecifikování útoku se pokouší vykonat všechny útoky, od nejjednodušších po složitější. Například při útoku na WPA se jedná o pořadí WPS Pixie-Dust, WPS PIN, PMKID, WPA Handshake. V tento moment ale není optimalizovaný v situacích, kdy daný útok na přístupový bod nefunguje. Teprve až po nějakém časovém intervalu se daný útok zruší a pokračuje se dalším, což zabere více času, než kdyby se daný útok rovnou zrušil. Jelikož v dnešní době jsou na většině přístupových bodů první tři útoky opraveny, zvyšuje se tak doba celkového útoku.

U útoků na zabezpečení WEP se jedná například chop-chop útok a další[5].

4.5 Hashcat

Hashcat je nástroj sloužící k prolamování hesel. Hlavní výhodou oproti již zmíněnému Aircrack-ng 4.1 je možnost provádění slovníkových útoků na grafických kartách. Tím se daný útok značně zrychluje. Další výhodou je například používání pravidel, podle kterých Hashcat může upravovat za běhu slovník, a vytvářet tak další kombinace.

Ukázka spuštění pro prolomení hesla ze zachyceného handshake v souboru *WPA.cap*:

- Hashacat používá jiný formát zachyceného .cap souboru a je nutné ho nejdříve převést na správný formát, např. pomocí nástroje *aircrack-ng* s použitím přepínače `-j`:
  `$ aircrack-ng WPA.cap -j converted`
- Následně můžeme použít vytvořený soubor *converted.hccapx* k prolomení hesla pomocí slovníkového útoku. Parametr `-m` s číslem 2500 značí, že se jedná o soubor, který obsahuje WPA-EAPOL obsah, cesta ke slovníku je zadána posledním parametrem.
  `$ hashcat -m 2500 converted.hccapx /slovniky/rockyou.txt`

Více informací a dokumentace se nachází na adrese[6].

4.6 Shrnutí

Výše popsané nástroje jsou již dlouho fungující a tradičně používané komunitou penetračního testování bezdrátových sítí. Kromě sady Aircrack-ng a Hashcat se jedná o automatizační skripty, které mají za cíl ulehčit uživateli daný útok. Na použití těchto nástrojů je přesto potřeba expertní znalost v dané problematice. K jejich použití také často není přesný scénář, a uživatel se musí rozhodovat sám, kdy např. konkrétní útok zrušit, protože už trvá neočekávaně dlouho. Kromě toho se u některých nástrojů vyskytují problémy s kompatibilitou (např. problém LAZY scriptu v určitých shellech viz odkaz[7]).

Během vytváření nástroje a testování již existujících nástrojů bylo zjištěno, že mnoho útoků na zabezpečení WPA je již opraveno i na starých routerech. Jedná se o WPS a PMKID útoky, viz 3.2. Na základě tohoto zjištění, částečně podloženém tabulkou routerů se zranitelností WPS (viz 3.3) a vlastními experimenty (kapitola 7.5), bylo rozhodnuto do nástroje zapracovat pouze slovníkové útoky v různých implementacích, jelikož proti tomuto útoku nemají routery téměř žádnou obranu. Do výsledného programu byly tedy vybrány nástroje ze sady Aircrack-ng a Hashcat. Vybrané nástroje samy nic neautomatizují a hodí se pro použití ve skriptování, což je pro tvorbu nového nástroje výhodné.

[5] https://github.com/derv82/Wifite2
[6] https://hashcat.net/wiki/
[7] https://github.com/arismelachroinos/lscript/issues/145

Kapitola 5

Obrana proti útokům

V dnešní době se bezdrátové sítě nacházejí v téměř všude. Motivací k útokům na tyto sítě může být mnoho, od získání přístupu k cizí síti až po skrývání identity při provádění nelegálních aktivit pomocí cizí sítě. Nejen z těchto důvodů je důležité vědět, jaké jsou možnosti obrany proti útokům na bezdrátové sítě.

Obecně platí pravidlo, že čím novější zabezpečení sítě použijeme, tím je bezpečnější. Například u nejstaršího zabezpečení WEP stačí na síti zachytit určitý počet inicializačních vektorů, ze kterých je následně možné heslo zjistit. Bylo ověřeno, že při použití 64bitového klíče se v průměru jedná o zhruba 35 000 vektorů, které je možné zachytit během několika desítek minut (experimenty byly prováděny na síti, kde jeden připojený uživatel sledoval videa z platformy *youtube.com*). Jelikož je tedy toto zabezpečení velmi snadno prolomitelné, je vhodné ho vůbec nepoužít. Na novějších zařízeních (např. telefonech s operačním systémem *Android 11*) je dnes uživatel dokonce varován během připojování k síti se zabezpečením WEP.

Nicméně i u novějších zabezpečení se nacházejí chyby, je možné zde zmínit například možnost použití WPS u sítí se zabezpečením WPA/WPA2. Zde při použití brute-force metody útočník prolamuje pouze maximálně 11 000 možných kombinací PIN, viz 3.3, což při použití moderních počítačů bude velmi rychlé. Útočník je zde limitován pouze časovým odstupem pro zkoušení nové kombinace na routeru. V případě, že by útočník byl schopen otestovat jeden PIN za 5 sekund, trval by mu útok maximálně 15,27 hodin. I když je dnes tato slabina na většině routerů opravena, je možné stále najít routery, na kterých by tento útok bylo možné provést. V případě, že není jisté, zdali daný router má slabinu opravenou, je vhodné možnost WPS vypnout.

Pro zjištění hesla sítě se zabezpečením WPA/WPA2 se nejčastěji používá slovníkový útok. Aby ho ale bylo možné provést, je nutné mít heš hesla, který je možné získat zachycením handshake, viz kapitola 3.2. Tuto metodu je možné zařadit do dvou kategorií, a to jak mezi aktivní, tak i mezi pasivní útoky. Rozdílem zde je, zdali útočník aktivně odpojuje uživatele s cílem zachytit handshake, nebo pouze pasivně čeká, než se k síti někdo připojí. V případě pasivní možnosti je jedinou možnou obranou proti tomuto útoku kvalita zvoleného hesla. Jelikož útočníkovi většinou není známá délka hesla, používá k útoku slovník, u kterého se domnívá, že by mohl obsahovat heslo zvolené k zabezpečení. V případě, že by délka hesla byla útočníkovi známá, naskytuje se možnost použití brute-force útoku, při kterém bude zkoušet všechny možné kombinace znakové sady.

Z následující tabulky (5.1) vyplývá, že je velmi důležité používat dlouhá hesla, která ideálně kombinují různé znaky (číslice, písmena - velká i malá a speciální znaky). Pro vytvoření takového hesla je možné použít různé generátory hesel, které jsou volně dostupné na

internetu např. generátor dostupný na adrese[1]. Použitím delších hesel je možné snížit pravděpodobnost prolomení pomocí brute-force útoku. V případě, že jsou použity znaky různých kategorií, snižuje se pravděpodobnost výskytu ve slovníku. Ukázkové heslo by mohlo být například: `OWud6@XUV1wgX@yG`, u kterého je téměř nemožné, aby se nacházelo ve vygenerovaném slovníku. Bohužel je ale velmi pravděpodobné, že většina uživatelů taková hesla nepoužívá kvůli jejich složitosti.

Délka hesla	[0-9]	[a-Z]	[a-Z] + [0-9]	[a-Z] + [0-9] + speciální znaky
6	Ihned	Ihned	Ihned	20 sekund
7	Ihned	2 sekundy	6 sekund	49 minut
8	Ihned	1 minuta	6 minut	5 dní
9	Ihned	1 hodina	6 hodin	2 roky
10	Ihned	3 dny	15 dní	330 let
11	Ihned	138 dní	3 roky	50 000 let
12	2 sekundy	20 let	162 let	8 000 000 let

Tabulka 5.1: Doba prolamování hesel v závislosti na jejich délce a znakové sadě

Výše zmíněná tabulka dostupná na stránce[2] počítá s využitím AWS p3.16xlarge (Amazon Web Services - cloudová služba nabízející výpočetní sílu). Při zmíněné konfiguraci je stroj vybaven následovně:

- 96x jader Intel Xeon Scalable (postaveno na architektuře Skylake) CPU
- 8x NVIDIA V100 Tensor Core GPU s pamětí 32 GB

S touto konfigurací je možné vyzkoušet za sekundu 632 miliard hesel, v době psaní této práce je cena za hodinu využití služby 25 USD, v přepočtu zhruba 535 Kč.

Pro lepší přehled o růstu výkonu v čase je možné použít nástroj dostupný na stránce[3], který umožňuje porovnání délky prolamování konkrétního hesla v různých letech pomocí brute-force metody. Například prolamování hesla `test1234` by v roce 2005 trvalo 6 dní a 4 hodiny (při testování 15288806.46 hesel za sekundu) V roce 2020 by se u stejného hesla jednalo o 1 den a necelých 22 hodin (při testování 17042497.3 hesel za sekundu). Při použití již zmíněné AWS by se v době psaní této práce jednalo zhruba o 6 minut.

Další možností je aktivní provedení útoku, přičemž se útočník snaží aktivně odpojit klienty, připojené k přístupovému bodu, pomocí deautentizačních rámců. Pokud se toto útočníkovi povede, odpojení klienti se pokusí opět připojit na síť, a útočník během jejich připojování může zachytit handshake. Deautentizační rámce je možné detekovat, a pomocí nich zjistit, zdali dochází k útoku. K jejich detekci je například možné použití programů typu IDS (Intrusion Detection System). Tyto programy zachytávají okolní síťový provoz a vyhodnocují možná rizika, jako například zasílání již zmíněných rámců atp. Konkrétním příkladem může být program *Wireless IDS (Intrusion Detection System)*. Dostupný je na otevřené platformě Github[4].

V případě, že detekce útoku nestačí, je možné mu zabránit použitím standardu 802.11w [3], který byl publikován v roce 2008. Tato nadstavba běžně používaného standardu IEEE

[1] https://www.lastpass.com/password-generator
[2] https://www.thesecurityfactory.be/password-cracking-speed/
[3] https://www.betterbuys.com/estimating-password-cracking-times/
[4] https://github.com/SYWorks/wireless-ids

802.11, přináší funkcionalitu Management Frame Protection (MFP), která umožňuje kontrolu integrity tzv. Management rámců, které jsou používány při deautentizaci [15]. Tato kontrola se v běžném standardu IEEE 802.11 nevyskytuje, a z toho důvodu je na něm možné útok provést.

Ačkoli bylo toto rozšíření představeno už v roce 2008, dodnes není ve většině běžných routerů dostupné. Další nevýhodou je, že MFP je dostupné pouze v routerech dražší cenové kategorie (např. *UBNT UniFi Dream Machine*, jehož cena se při vytváření práce pohybuje kolem 8000 Kč). Toto bezpečnostní opatření je již zahrnuto v novém zabezpečení WPA3, které se postupně začíná používat, a tak by v budoucnu měl být tento útok vyřešen.

Kapitola 6

Generování slovníků

Kvalita slovníku zásadně ovlivňuje dobu a pravděpodobnost úspěšnosti prolamování. Kvalitní slovník je důležitý zejména kvůli přihlédnutí k ostatním možnostem prolamování, popsaných v kapitole 3.

Slovník by měl obsahovat hesla, která by daný uživatel mohl mít s větší pravděpodobností zvolena. Na internetu nebo v Linuxových distribucích, sloužících k penetračnímu testování, je možné najít ukázkové slovníky, které je možné použít. Tyto slovníky se většinou skládají z databáze uniklých hesel. Jejich výhodou je, že jsou většinou seřazeny podle frekvence užívání hesel (na začátku jsou častěji používaná hesla na základě uniklých databází), nevýhodou ale je, že jsou orientovány zejména na anglicky mluvící země. Tím se tedy snižuje pravděpodobnost nálezu hesla při použití v českém prostředí.

Česká hesla mají řadu specifik a nemusí být obsažena ve slovnících pro obecné použití. Z toho důvodu je výhodné mít i slovník čistě českých hesel. Po prohledání webu a darkwebu nebyly nalezeny žádně čistě české slovníky, nejvíce se podobajícím nálezem byl seznam českých slov (dostupný na stránce[1]). Samotný takový slovník by k prolamování hesel nebylo vhodné použít, a proto se tato práce dále zabývá možnostmi vytváření vlastních slovníků.

6.1 Získávání dat

V případě, že se rozhodneme vytvořit vlastní slovník, je ve většině případů nutné mít data, ze kterých se slovník bude vytvářet. Zde tedy vyvstávají dvě otázky, jaká data vlastně hledat a kde tato data získat.

V případě vytváření slovníku pro útok na konkrétní osobu by bylo možné vzít v úvahu slova, která souvisejí např. s koníčky dané osoby, nebo obecně s prostředím, ve kterém se osoba vyskytuje. V tomto případě je ale velmi náročné nasbírat takové množství dat o konkrétní osobě, aby byla personalizace pro tvorbu hesel použitelná. Při vytváření slovníku pro obecné použití v Česku je tato otázka ještě složitější. V úvahu zde připadají všechna česká slova, vlastní jména, názvy ulic, sportovních klubů atp.

Tímto způsobem však získáme obrovský slovník z něhož jen mizivé procento by bylo použito jako heslo. Pro získání všech českých slov a vlastních jmen je možné nalézt volně dostupná data na internetu (např. seznam českých slov[2]). Obdobně je možné získat data i pro další již zmíněné návrhy (např. názvy ulic).

[1] http://www.zip-password-cracker.com/dictionaries.html
[2] http://www.jmenaprijmeni.cz/seznam-ceskych-slov

Další možností je použití nástroje CeWL - Custom Word List generator dostupného na adrese[3]. Cílem tohoto nástroje je automatizovaný sběr slov ze zadané webové stránky. Při výchozím spuštění se nástroj pouští do hloubky 2 od zadaného url, nicméně toto nastavení je možné pomocí argumentu změnit.

- Ukázka použití nástroje CeWL na webové stránce www.seznam.cz, parametr `-d` určuje hloubku zanoření, `-w` určuje název vygenerovaného slovníku s cestou
  `$ ruby -W0 ./cewl.rb https://www.seznam.cz/ -d 1 -w ./dict.txt`
- Ukázka výsledného souboru:

```
Před
Skrýt
Komentáře
dny
pro
ale
Seznam
které
streaming
hodinami
```

Při tomto konkrétním spuštění zapsal nástroj celkem 14618 unikátních slov.

6.2 Generování slovníku ze získaných dat

Po získání dat, je možné vygenerovat slovník, jedná se ale o stále otevřený problém. Můžeme zde vycházet z typických hesel končících posloupností čísel nebo slov začínajících velkým písmenem, dále je možné měnit podobná písmena na číslice atd. Každopádně nikdy si nemůžeme být úplně jistí, že právě daná úprava je správná.

Předpokládejme, že máme textový soubor obsahující slova, získaný například nástrojem CeWL, případně by se zde dal použít i neúplný slovník českých slov zmíněný na začátku této kapitoly. Nyní bychom si mohli napsat vlastní script, který by vstupní data upravoval dle našich představ, nicméně to v dnešní době není nutné, jelikož existuje už několik nástrojů k tomu určených.

Hashcat

Nástroj Hashcat je kromě prolamování hesel možné použít i ke generování slovníků. Ke specifikování výsledného slovníku se zde používají tzv. pravidla, kterými definujeme, co se se vstupními daty má stát. Tato pravidla se zapisují do oddělených souborů a jejich výhodou je, že mají stejnou syntaxi jako pravidla pro další nástroj John the Ripper. Pravidla se dají použít i rovnou během provádění útoku. Jejich kompletní přehled je možné nalézt na odkaze[4].

[3] https://github.com/digininja/CeWL
[4] https://hashcat.net/wiki/doku.php?id=rule_based_attack

- Ukázka použití generování slovníku pomocí dvou pravidel:
  - Soubor *append.rule* s pravidly pro dopsání znaků na konec slova:

    `:` – neprovedení žádné akce (slovo zůstane stejné)

    `$1` – přidání znaku 1 na konec slova

    `$2` – přidání znaku 2 na konec slova
  - Soubor *cap.rule* s pravidly pro zvětšení prvního písmene:

    `:`

    `c` – zvětšení prvního písmene
  - Obsah vstupního souboru *test.txt*:

    `test`

- Spuštění nástroje s oběma pravidly:

  ```
  $ hashcat -r append.rule -r cap.rule test.txt --stdout > vystup.txt
  ```

- Vytvořený slovník:

  ```
  test
  test1
  test2
  Test
  Test1
  Test2
  ```

John the Ripper

Dalším z nástrojů pro generování slovníků ze zadaných vstupních dat je John the Ripper dostupný na adrese[5]. Jeho funkcionalita na základě pravidel je velmi podobná jako u předchozího nástroje, proto je zde syntaxe pravidel ekvivalentní. Jediným rozdílem je, že pravidla se musí vložit do speciálního souboru, který je uložen v `/etc/john/john.conf`[6] – může se lišit pro konkrétní distribuci.

- Ukázka spuštění nástroje:

  ```
  $ john --wordlist=test.txt --rules --stdout > vystup.txt
  ```

Mentalist

Mentalist funguje opět velmi podobně, ale jeho výhodou je grafické uživatelské rozhraní, pomocí kterého je možné zadávat jednotlivá pravidla. To se může hodit zejména méně zkušeným uživatelům, jelikož zde nemusí rozumět syntaxi pravidel, která je opět ekvivalentní s již výše zmíněnými. Nástroj je možné nalézt na stránce[7].

[5] https://www.openwall.com/john/

[6] https://netsec.ws/?p=457

[7] https://github.com/sc0tfree/mentalist

Další nástroje

Výše zmíněné nástroje počítají se vstupními daty, která na základě pravidel upravují a vytváří tak větší slovníky. Existují ale i další nástroje, které dokáží slovník vygenerovat samy. Hodí se ale spíše pro případy, kdy víme, jak dlouhé je dané heslo a z jakých znaků je složené. Jedná se například o nástroj **Crunch**[8].

Pro případ, kdy známe informace o majiteli sítě, existuje nástroj **CUPP – Common User Passwords Profiler**[9]. Pomocí tohoto programu je možné zadat známé informace, jako např. jméno a příjmení majitele, datum narození, přezdívku atd. Na základě těchto dat nástroj vygeneruje slovník založený na těchto informacích. Podobným nástrojem je i **Pydictor**[10], pomocí kterého je ale možné i generování slovníku obdobně jako u nástroje **Crunch**.

6.3 Měření účinnosti vytvořeného slovníku

Náhodným sestavováním hesel z jednotlivých fragmentů vznikají používaná hesla jen s malou pravděpodobností. To jestli konkrétní vytvořené heslo bylo někdy použito nelze ověřit. V tento moment by bylo ideální provést pokus prolamování běžně používaných Wi-Fi sítí a následně při větším vzorku odhadnout, jestli je vytvořený slovník dostatečný. Tato metoda ověření ale bohužel není možná, jelikož je tato aktivita nelegální. Pro přibližný odhad efektivity metody pro vytváření slovníku byla použita metrika popsaná níže.

Bylo by možné použít databázi uniklých hesel ze stránky[11], která obsahuje přes 600 milionů hesel a porovnat je s hesly z vygenerovaného slovníku. Zde ale narážíme na více problémů, databáze uniklých hesel opět není čistě pro české prostředí, a hesla jsou zde zahešovaná pomocí SHA-1, je zde tedy nutné vytvořený slovník zahešovat a porovnání provádět až následně. Větším problémem je ale velikost této databáze a náročnost výpočtu shody. Jelikož databáze má zhruba 29 GB, nebylo možné provést rychlé porovnání na běžném notebooku s RAM 8 GB. Konkrétní výpočty v tabulce 6.1 tedy počítají shodu s referenčním slovníkem *rockyou.txt*, který je možné nalézt na odkazu[12]. Slovník *rockyou.txt* také obsahuje již uniklá hesla, oproti zmíněné databázi je ale mnohem menší (obsahuje pouze 14 344 392 hesel) a není zahešovaný.

Měření bylo provedeno následovně:

1. Získání dat pro vytvoření slovníku
2. Výpočet shody původních dat s referenčním slovníkem
3. Vygenerování slovníku ze získaných dat
4. Výpočet shody vygenerovaného slovníku s referenčním slovníkem
5. Výpočet zlepšení Z: $Z = N/P$; $N = S_N/D_N$; $P = S_P/D_P$, kde:
   P je poměr shody získaných dat k jejich délce,
   N je poměr shody vygenerovaného slovníku k jeho délce,
   S je počet shodujících se hesel mezi slovníkem a databází,

[8] https://tools.kali.org/password-attacks/crunch
[9] https://github.com/Mebus/cupp
[10] https://github.com/LandGrey/pydictor
[11] https://haveibeenpwned.com/Passwords
[12] https://github.com/brannondorsey/naive-hashcat/releases/download/data/rockyou.txt

D je počet slov slovníku (popř. získaných dat),
Z je poměr zlepšení zohledňující zvětšení slovníku při zvýšení shody

V tabulce 6.1 jsou uvedeny tři příklady tvorby slovníků s vypočtenými hodnotami zlepšení. Použit byl neúplný český slovník (zmíněný v prvním odstavci), slovník vygenerovaný nástrojem CeWL ze stránky *seznam.cz* (stejný jak v ukázce nástroje CeWL) a pro srovnání i anglický slovník dostupný na adrese[13]. Proces úpravy slovníků přidává na konec slov čísla: `1, 12, 123, 1234` a opakuje totéž se zvětšením prvního písmene slova.

Jelikož slovník *rockyou.txt* není opět pro české prostředí, byl zde pro srovnání přidán i slovník anglický. Může se zdát překvapivé, že zlepšení anglického slovníku je mnohem menší než zlepšení českých slovníků. Je tomu tak z důvodu, že už v neupraveném anglickém slovníku se nachází velký počet shodujících se slov, nicméně se zde upravením slovníku dostáváme na nejvyšší shodu.

Slovník	**Počet slov**	**Shoda**	**Nárůst shody**	**Zlepšení**
Český	272058	8644	-	-
Český - upravený	2448522	220871	25.55x	2.8391x
Seznam.cz	14618	911	-	-
Seznam.cz - upravený	131562	20996	23.05x	2.5608x
Anglický	466550	61914	-	-
Anglický - upravený	4198950	404503	6.53x	0.7259x

Tabulka 6.1: Měření zlepšení při tvorbě konkrétních slovníků

[13] https://github.com/dwyl/english-words/blob/master/words.txt

Kapitola 7

Implementace nástroje a testování

Projekt FlexProbe, který realizuje výzkumná skupina ANT (skupina akcelerovaných síťových technologií) na VUT FIT, má za cíl vytvoření sondy, která by měla sloužit k zákonným odposlechům. Využívat by jej následně měla Policie ČR, která by díky této sondě mohla mít možnost za daných okolností odposlouchávat provoz na síti konkrétních uživatelů, a na základě analýzy provozu vyhodnocovat nelegální aktivitu.

Aby bylo možné provoz na síti analyzovat a zjišťovat z něj informace, je nutné mít přístup k nezašifrované komunikaci. Nástroj implementovaný v této práci umožňuje zjistit heslo ke konkrétní bezdrátové síti, pomocí něhož je následně možné danou komunikaci dešifrovat.

7.1 Popis nástroje

V kapitole 4 jsou popsány některé z již existujících nástrojů, které slouží k podobnému účelu, nicméně nejsou vždy uživatelsky přívětivé, a to zejména pro nezkušené uživatele.

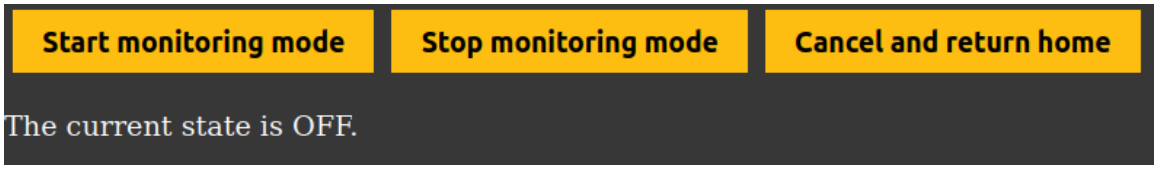
Proto jedním z kritérií výsledného nástroje byla jednoduchost jeho použití. Vytvořený nástroj je tedy plně automatizovaný a doplněný o REST API, díky kterému je možné jej ovládat z webového rozhraní (viz 7.2) a integrovat jej do dalších systémů.

Vytvořený nástroj tedy umožňuje provedení útoku na zvolenou síť, a měl by být zařazen jako součást projektu FlexProbe a následně zahrnut do platformy popsané níže (viz 7.4). Výhodou nástroje je, že je možné ho použít nejen z této platformy, ale také z jakéhokoli dalšího stroje, vybaveného operačním systémem Linux se síťovou kartou, která umožňuje injekci paketů a monitorovací režim.

7.2 Použití nástroje

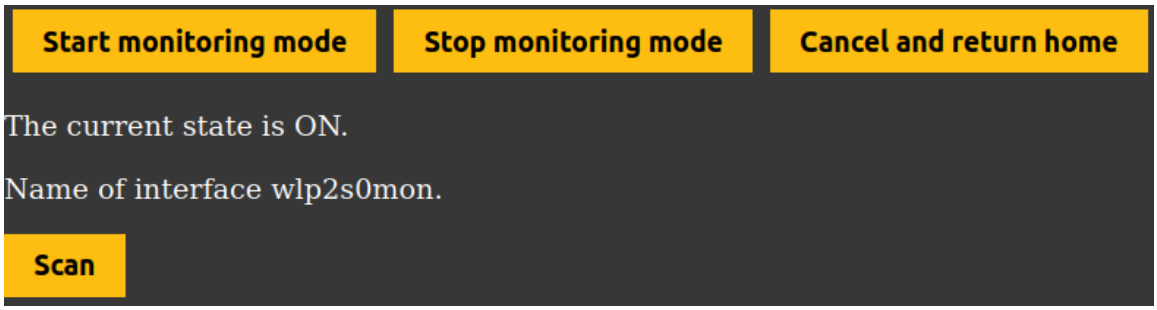
Následující scénář popisuje typické použití nástroje. Na začátku uživatel specifikuje síť, ke které chce zjistit heslo, a ke které má povolení k provedení útoku (ať už na základě soudního příkazu, nebo povolení od majitele sítě – např. pro kontrolu kvality zabezpečení).

1. Uživatel spustí nástroj a ve webovém prohlížeči si otevře příslušnou IP adresu, na které je webové rozhraní (IP adresu ručně zadá při spuštění programu).



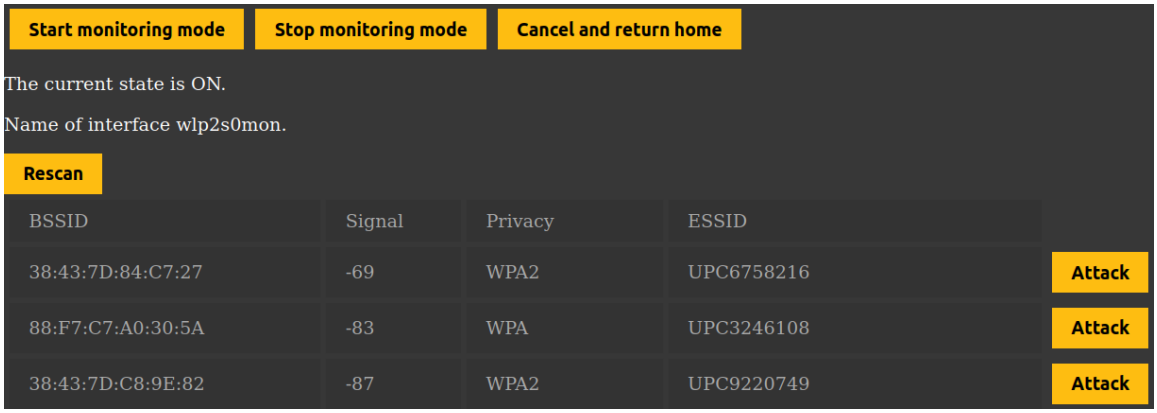
Obrázek 7.1: Výchozí stav aplikace

2. Pomocí tlačítka přepne síťovou kartu do monitorovacího režimu.



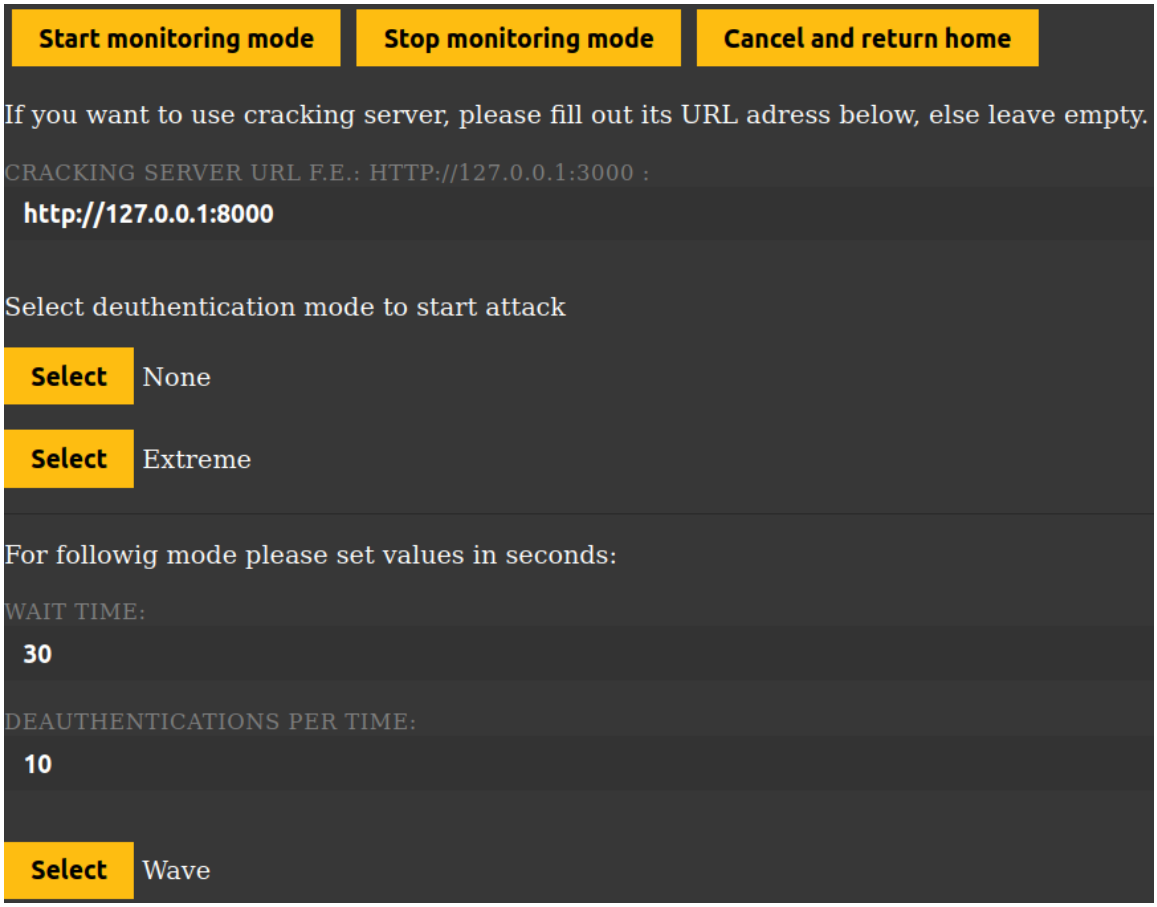
Obrázek 7.2: Stav aplikace po zapnutí monitorovacího režimu

3. Stisknutím tlačítka zahájí skenování dostupných Wi-Fi sítí.
4. Z tabulky dostupných sítí vybere danou síť a zahájí útok.



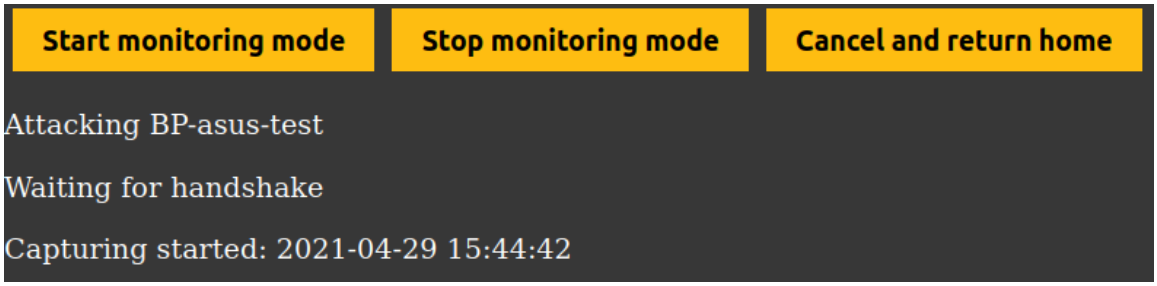
Obrázek 7.3: Seznam dostupných sítí

- V případě, že se jedná o síť s WEP zabezpečením, začnou se rovnou zachytávat inicializační vektory, které slouží k prolomení hesla.
- V případě zabezpečení WPA vybere uživatel stupeň agresivity pro získání handshake, a případně zadá adresu crackovacího serveru (viz 7.3).



Obrázek 7.4: Volba módu získání handshake

5. Po zahájení útoku uživatel už jen čeká na dokončení útoku.



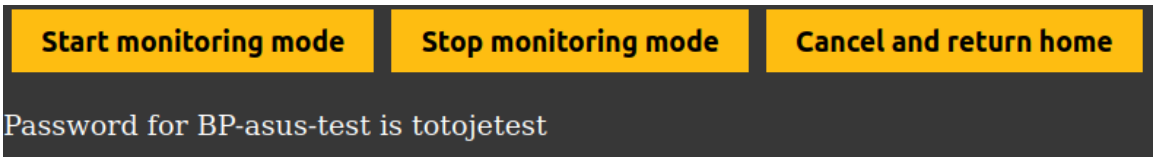
Obrázek 7.5: Stav probíhajícího útoku

- V případě WEP zabezpečení je nutné zachytit určitý počet inicializačních vektorů. Při experimentech bylo zjištěno, že například pro 64bitový klíč se v průměru jedná o cca 35 000 inicializačních vektorů k úspěšnému prolomení. Doba trvání zachytávání závisí na počtu připojených uživatelů a také na provozu na síti. Může se jednat o několik minut až po několik hodin. V momentě, kdy na síti není nikdo připojen, není možné zachytávat inicializační vektory.
- V případě WPA zabezpečení je nutné na místě zachytit handshake. Zde doba trvání zachycení závisí na tom, jestli jsou na síti připojeni uživatelé, a pokud

ano, tak také na zvolení agresivity útoku. Handshake je možné zachytit jen na síti, kde je někdo připojen, nebo na síti, na kterou se během zachytávání zrovna někdo připojí.

Pro následné prolamování hesla už uživatel nemusí být v dosahu sítě, a doba trvání se odvíjí od délky použitých slovníků a také od hardware zařízení, na kterém se prolamování provádí. Může trvat od několika minut po několik hodin až dní.

6. V případě úspěchu se uživateli zobrazí heslo pro danou síť Wi-Fi.

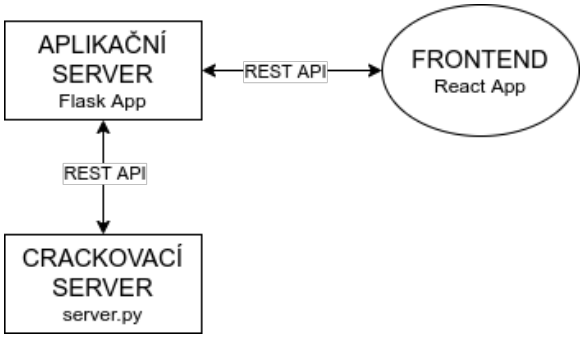


Obrázek 7.6: Získané heslo

Podrobněji je běh programu popsán v obrázku 7.8.

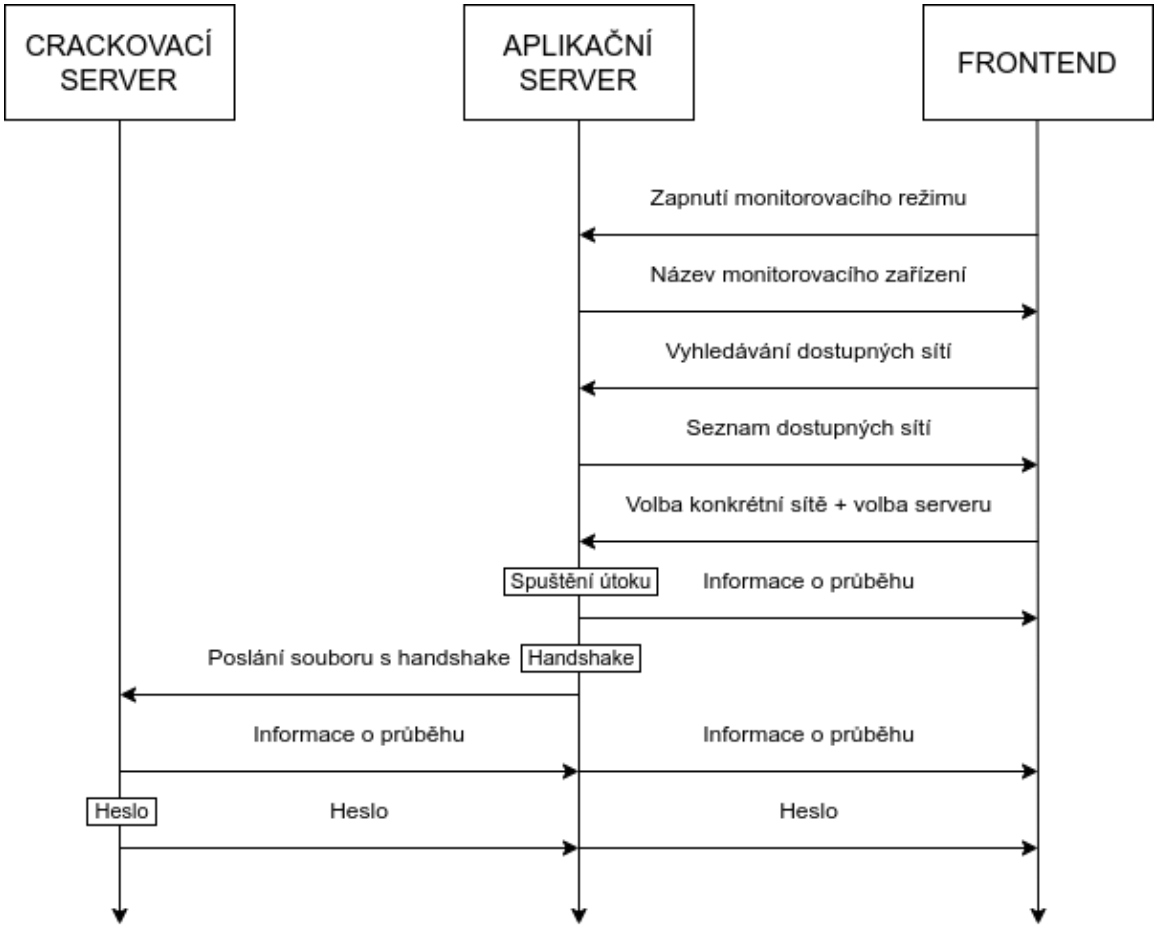
7.3 Součásti nástroje

Implementace celého nástroje se skládá z několika modulů, logicky je ale můžeme rozdělit na tři části. Jedná se o aplikační server, frontend a crackovací server. Přičemž crackovací server je modul navíc, který k funkčnosti celého nástroje není nutný. Je možné ho však využít v případě spuštění se sondou. Propojení všech částí je naznačeno na obrázku 7.7.



Obrázek 7.7: Diagram propojení

Při spuštění aplikace se spustí webové rozhraní na zadané IP adrese a portu. Uživatel následně všechny činnosti provádí z rozhraní, kde se všechny aktivace tlačítek a vyplňování formulářů přenášejí do aplikačního serveru jako POST dotazy na dané url umožňující přijímání http dotazů. V aplikačním serveru se pomocí aktivace konkrétních url spouští vybrané funkce a následně se posílají data opět do webového rozhraní. Aplikační server tedy funguje zároveň jako API. V případě využití crackovacího serveru přeposílá API soubor se zachyceným handshake na crackovací server a následně s ním komunikuje, aby mohl posílat informace o stavu prolamování hesla do rozhraní. Diagram ukázkové komunikace je zobrazen na následujícím obrázku (7.8).



Obrázek 7.8: Ukázka komunikace při útoku na WPA s použitím crackovacího serveru

Aplikační server

Aplikační server a zároveň hlavní část programu obsahuje několik modulů (*attacks.py*, *tools.py*, *killableThread.py* a *web.py*), které slouží jednak k provádění útoků na Wi-Fi sítě, prolamování hesel, ale i jako API pro webové rozhraní. Všechny tyto moduly jsou napsané v jazyce Python3.

Základem celého aplikačního serveru je modul *tools.py*, jenž obsahuje funkce sloužící jak k přípravě zařízení pro útok, tak veškerému provádění útoků a získávání informací o právě probíhajících útocích. Tyto funkce jsou v rámci celého nástroje volány z dalších modulů. Například v modulu *attacks.py* jsou útoky složeny právě z těchto funkcí, dále se používají v modulu pro webové API, např. pro zapnutí monitorovacího režimu, nebo pro skenování dostupných sítí. Funkce z tohoto modulu jsou využívány i crackovacím serverem. Externí nástroje (*Aircrack-ng* a *Hashcat*) se zde spouští pomocí knihoven `subprocess` a `Pexpect`.

Modul *attacks.py* obsahuje funkce, které slouží k provedení útoku na sítě s WPA zabezpečením, jelikož je komplikovanější než útok na sítě se zabezpečením WEP, kde se jedná pouze o spuštění daného externího nástroje s určitými parametry.

Modul *killableThread.py* obsahuje upravený konstruktor třídy `Thread` z knihovny `threading`. Díky této úpravě je možné libovolně ukončovat pracující vlákna, což při stan-

dardní implementaci Threadů v jazyce Python není možné. Implementace modulu vychází z článku dostupného na adrese[1].

Modul *web.py* tvoří hlavní aplikační část, která po zvolení přístupového bodu rozhodne, na základě použitého zabezpečení, který útok se má provést a zároveň je hlavním vláknem pro jeho provádění. Dalším jeho úkolem je propojení s webovým rozhraním, přičemž pracuje jako API a je postaven za pomoci Flasku – jedná se o mikrowebový framework napsaný v jazyce Python.

Flask zde pomocí dekorátorů přiděluje funkcím modulu konkrétní url (endpoint), které slouží k aktivaci funkcí aplikačního serveru z webového rozhraní pomocí http dotazů. Dále je lze ale použít i k poskytování informací o průběhu útoku pomocí periodických dotazů.

Součástí tohoto API je i funkcionalita spojení s crackovacím serverem, který je rovněž postaven za pomoci Flasku.

Frontend

Frontend celého nástroje je implementován pomocí knihovny React[2] v jazyce JavaScript (ECMAScript 6.0 a JSX). Ovládací rozhraní se nachází na jedné webové stránce, která se skládá z více React komponent a využívá kaskádové styly používané v celém projektu FlexProbe.

Pro překlad se využívá nástroj *webpack* s konfigurací pro balíčkovací manažer *npm*, pomocí kterého je možné provést build. V rámci buildu se vytvoří soubor html, který zahrnuje minifikovanou aplikaci, zařizující vykreslování obsahu stránky.

Hlavní komponenta App, implementovaná funkcí v souboru *App.js*, dokáže na základě svého stavu vykreslovat konkrétní, v daný moment chtěný, obsah. Tento stav je synchronizovaný se stavem z API. V případě útoků se informace o průběhu zobrazují pomocí komponent Wpa a Ivs, které jsou implementovány opět pomocí funkcí v souborech *wpa.js* a *ivs.js*. Komponenty se periodicky dotazují aplikačního serveru a zjišťují tak stav útoku. Předchozí dotazy se provádějí do doby, než je heslo prolomeno a následně v rozhraní zobrazeno.

Celý frontend byl záměrně implementován jako jedna React aplikace. V aplikaci se tedy nenachází žádná přesměrování, a veškeré změny stavů se zajišťují pomocí REST API. Díky tomu je aplikaci možné pohodlně zařadit do dalších stránek při integraci do webového rozhraní projektu FlexProbe.

Crackovací server

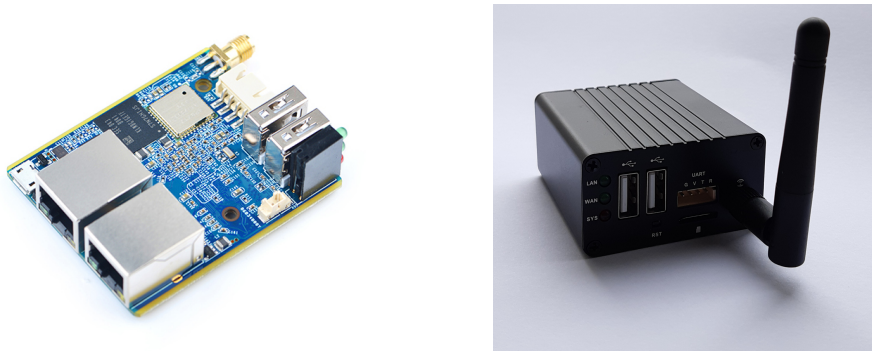
V projektu FlexProbe se počítá s využíváním sondy, která bude pravděpodobně napájená baterií. Kromě toho i kvůli výkonu sondy není vhodné, aby se prolamování hesel provádělo na ní. A proto byl pro tento účel implementován crackovací server, na který je možné tuto úlohu delegovat.

Prolamování na crackovacím serveru probíhá úplně stejně, jako kdyby se provádělo lokálně například z notebooku. S tím rozdílem, že pakety se zachyceným handshake se zašlou na zadanou adresu crackovacího serveru a prolamování hesla probíhá tam. Pomocí přesměrování přes API se informace o prolamování zobrazují ve webovém rozhraní. Výhodou je, že crackovací server může být nainstalovaný na výkonném počítači, který může být vybaven např. několika grafickými kartami, prolamování tedy bude značně rychlejší. Jedná se opět o modul napsaný v jazyce Python3 s využitím frameworku Flask velmi podobně, jak

[1] https://www.geeksforgeeks.org/python-different-ways-to-kill-a-thread/
[2] https://reactjs.org/

v již zmíněném modulu *web.py*. Server je vybavený frontou procesů, a tak je možné na něj delegovat úlohy z více sond.

7.4 Platforma Nanopi R1



Obrázek 7.9: Sonda Nanopi R1

V rámci projektu FlexProbe je na základě svých parametrů použita platforma Nanopi R1. V malém balení poskytuje potřebný hardware pro odposlechy za nízkou cenu. Jediným HW nedostatkem je síťová karta, která nepodporuje monitorovací režim, nicméně to ale není zásadní problém, jelikož do sondy je možné připojit externí síťovou kartu pomocí USB rozhraní. Pro potřeby tohoto projektu je totiž nutné mít síťové karty dvě, přičemž jedna slouží ke komunikaci a druhá je použitá na provádění útoků. Při využití platformy získává uživatel možnost umístit ji na nepozorované místo, a následně může program ovládat z méně nápadného místa. Díky tomu je možné nástroj použít bez vzbuzení pozornosti, což je ve většině daných situací, zejména pro použití při legálních odposleších, vhodné.

Zvolená platforma má následující parametry dostupné ze stránky[3]:

- CPU: Allwinner H3, Quad-core Cortex-A7 s až 1.2 GHz
- RAM: 512 MB/1 GB DDR3
- 1G Ethernet, 100M Ethernet
- Wi-Fi: 802.11b/g/n, s SMA konektorem
- MicroSD, UART
- Velikost: 50.5 x 60 mm
- OS: Ubuntu, OpenWRT

Platforma pracuje na operačním systému Linux, a tak se od běžného počítače téměř neliší. Rozdíly je samozřejmě možné nalézt v hardwarové stránce. Sonda využívá ARM procesor, vyznačující se určitými specifikacemi, které mohou komplikovat podporu nových nástrojů. Pro požadavky projektu FlexProbe je ale dostačující. Nástroj se tedy dá do sondy pohodlně integrovat. Součástí výsledného nástroje je i již zmíněný crackovací server, který je možné spustit na výkonném počítači a prolamování provádět na něm viz 7.3.

[3] http://wiki.friendlyarm.com/wiki/index.php/NanoPi_R1#Hardware_Spec

Nástroj byl na sondě spuštěn v operačním systému Armbian Buster s kernelem 5.10.y (dostupným z adresy[4]). Jediný problém při instalaci na sondu nastal s chybějícím balíčkem *Hashcat*, který ale není pro sondu nijak důležitým, jelikož se používá pro prolamování hesel za pomoci grafických karet.

7.5 Testování

Pro tvorbu automatizovaného prolamovače Wi-Fi zabezpečení bylo nejdříve nutné ozkoušet již existující nástroje z kapitoly 4. Jelikož jsou útoky na cizí sítě bez povolení majitele nelegální, byly v rámci testování experimenty prováděny na různých routerech v domácí síti.

Pro testování útoků na zabezpečení WPA/WPA2 bylo dostačující provádět testování na routeru bez přístupu k internetu, jelikož se jednalo zejména o zachycení handshake. U zabezpečení WEP však bylo žádoucí mít síť s připojením k internetu. Proto při těchto útocích byly sítě plně funkční, a byl na nich připojen uživatel, který pro generování provozu sledoval videa ze služby *youtube.com*. Pro sítě byla nastavována hesla různých složitostí.

Z vybraných nástrojů byl následně sestaven výsledný automatizační nástroj, který byl testován následovně:

1. Spuštění automatizovaného útoku na danou síť
2. Využití druhého notebooku/telefonu jako klientů sítě z důvodů:
   - Testování deautentizačních útoků
   - Generování provozu
   - Simulování skutečných uživatelů
3. Po dokončení útoků bylo ověřeno, zdali se heslo shoduje

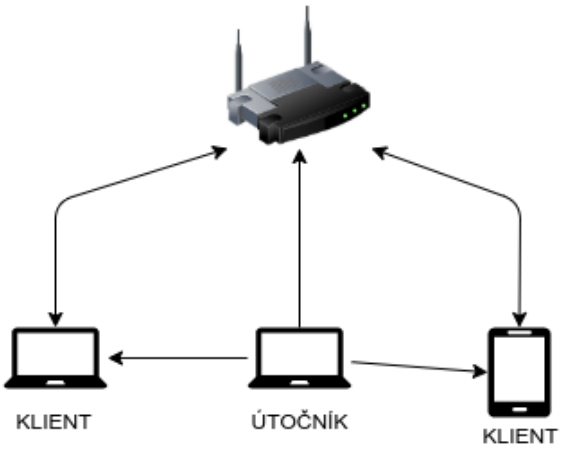
Testování probíhalo za pomoci dvou notebooků vybavených síťovými kartami `Intel Corporation Wireless 3165(rev 79)` a `Intel Corporation Wireless 3160(rev 83)` a telefonem *Samsung A40*, schéma testovací sestavy je na obrázku 7.10.

K testování byly využity následující routery:

- D-Link DI-524
- Edimax BR-6204WLg
- Tenda N3
- Comtrend VR-3031eu
- Asus RT-N16
- Asus DSL-AC52U
- Compal CH7465LG
- Mobilní hotspot (Samsung A40, Android 11)

Přehled jejich zabezpečení a útoků, které je možné na dané routery použít, se nachází v tabulkách 7.1 a 7.2 na následující straně.

[4] https://www.armbian.com/nanopi-r1/



Obrázek 7.10: Diagram propojení

Router	WEP	WPA	WPA2	WPS
D-Link DI-524	Ano	Ano	Ano	Ne
Edimax BR-6204WLg	Ano	Ano	Ano	Ne
Tenda N3	Ne	Ano	Ano	Ne
Comtrend VR-3031eu	Ne	Ano	Ano	Ano
Asus RT-N16	Ne	Ano	Ano	Ano
Asus DSL-AC52U	Ne	Ano	Ano	Ano
Compal CH7465LG	Ne	Ano	Ano	Ano
Mobilní hotspot	Ne	Ne	Ano	Ne

Tabulka 7.1: Dostupná zabezpečení testovaných routerů

Router	WEP útok	Deautentizace	PMKID	WPS
D-Link DI-524	Ano	Ano	Ne	Ne
Edimax BR-6204WLg	Ano	Ano	Ne	Ne
Tenda N3	-	Ano	Ne	Ne
Comtrend VR-3031eu	-	Ano	Ne	Ne
Asus RT-N16	-	Ano	Ne	Ne
Asus DSL-AC52U	-	Ano	Ne	Ne
Compal CH7465LG	-	Ne	Ne	Ne
Mobilní hotspot	-	Ano	-	-

Tabulka 7.2: Proveditelnost útoků pro zmíněné routery

Kapitola 8

Závěr

Tato práce se věnuje problematice prolomení Wi-Fi zabezpečení a následnému získání hesla k dané síti. Data se po síti posílají zašifrovaná, a aby bylo možné zachycená data analyzovat, je nutné dostat se k nim v dešifrované podobě, čemuž se zabezpečení sítí z pochopitelných důvodů snaží zabránit. Hlavním cílem práce je vytvoření nástroje, který umožňuje prolomení hesel Wi-Fi sítí, pomocí kterých je možné zachycená data dešifrovat. Nástroj by měl být následně využit v rámci projektu FlexProbe, který vytváří sondu pro zákonné odposlechy a obstarává právě analýzu odposlouchávaných dat.

Pro vytvoření nástroje bylo nejdříve nutné prostudovat a zanalyzovat dostupná zabezpečení Wi-Fi sítí a jejich fungování. Této analýze se věnuje kapitola 2. Následně bylo nutné zjistit, jaké existují možnosti pro získání nebo prolomení hesla k různým zabezpečením bezdrátových sítí. Kapitola 3 tedy pojednává o existujících útocích na konkrétní typy zabezpečení.

Po nastudování těchto informací bylo možné zaměřit se na vyhledání již existujících nástrojů sloužících k podobnému účelu a provedení jejich testování (viz kapitola 4). Na základě provedených experimentů bylo zjištěno, že útoky cílené na slabiny WPS a PMKID u zabezpečení WPA/WPA2 byly na většině zařízení velmi rychle opraveny. Z tohoto důvodu byl do výsledného nástroje zařazen pouze útok na zachycení handshake, ze kterého je možné následně pomocí slovníkového útoku heslo zjistit. Kromě tohoto útoku nástroj obsahuje i útok na sítě se zabezpečením WEP.

Z důvodu omezení útoků na zabezpečení z rodiny WPA bylo nutné věnovat se v práci i získávání nebo generování slovníků k provedení slovníkového útoku. Jak již bylo zmíněno v kapitole 6, nepodařilo se nalézt slovníky uniklých českých hesel. V kapitole 6 jsou tedy shrnuty způsoby získávání dat pro tvorbu slovníků a možnosti jejich vytváření. Dále je zde navržená jednoduchá metrika pro zjištění zlepšení vytvořeného slovníku oproti původním získaným datům. V případě, že by se v budoucnu objevil slovník s uniklými českými hesly, bylo by možné na základě jeho analýzy vytvořit ideální slovník, což by vedlo ke zefektivnění crackování při použití slovníkových útoků.

Jelikož se v projektu FlexProbe počítá zejména s využitím sondy, bylo žádoucí pro nástroj vytvořit webové rozhraní, pomocí kterého bude možné nástroj ovládat. Rozhraní bylo tedy cíleně implementováno jako jedna React aplikace, kterou bude při zařazení do projektu jednoduché integrovat. Další částí práce bylo vytvoření crackovacího serveru, jelikož prolamování hesel je výpočetně náročné, a zvolená sonda nedisponuje parametry, které by k tomu byly vhodné.

Do budoucna je plánováno rozšířit tento nástroj o modul s útoky na WPA3. Tento modul v současnosti vzniká v rámci jiné práce. V době psaní této práce však zatím nebyly

nalezeny použitelné nástroje. Integrace modulu pro WPA3 bude v blízké budoucnosti nutná vzhledem k předpokládanému rozšiřování WPA3.

Literatura

[1] IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. *IEEE Std 802.11-1997*. 1997, s. 1–445. DOI: 10.1109/IEEESTD.1997.85951.

[2] IEEE Standard for information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 6: Medium Access Control (MAC) Security Enhancements. *IEEE Std 802.11i-2004*. 2004, s. 1–190. DOI: 10.1109/IEEESTD.2004.94585.

[3] IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 4: Protected Management Frames. *IEEE Std 802.11w-2009 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, and IEEE Std 802.11y-2008)*. 2009, s. 1–111. DOI: 10.1109/IEEESTD.2009.5278657.

[4] IEEE Standard for Information Technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 10: Mesh Networking. *IEEE Std 802.11s-2011 (Amendment to IEEE Std 802.11-2007 as amended by IEEE 802.11k-2008, IEEE 802.11r-2008, IEEE 802.11y-2008, IEEE 802.11w-2009, IEEE 802.11n-2009, IEEE 802.11p-2010, IEEE 802.11z-2010, IEEE 802.11v-2011, and IEEE 802.11u-2011)*. 2011, s. 1–372. DOI: 10.1109/IEEESTD.2011.6018236.

[5] IEEE Standard for Information technology—Telecommunications and information exchange between systems Local and metropolitan area networks—Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)*. 2016, s. 1–3534. DOI: 10.1109/IEEESTD.2016.7786995.

[6] Adnan, A. H., Abdirazak, M., Sadi, A. S., Anam, T., Khan, S. Z. et al. A comparative study of WLAN security protocols: WPA, WPA2. In: *2015 International Conference on Advances in Electrical Engineering (ICAEE)*. 2015, s. 165–169. DOI: 10.1109/ICAEE.2015.7506822.

[7] Barken, L. *How Secure is Your Wireless Network? Safeguarding Your Wi-Fi LAN*. 1. vyd. Pearson, srpen 2003. ISBN 0-13-140206-4.

[8] Bongard, D. Offline bruteforce attack on WiFi Protected Setup. 2014, [cit. 2021-1-27]. Dostupné z: http://archive.hack.lu/2014/Hacklu2014_offline_bruteforce_attack_on_wps.pdf.

[9] Caneill, M. a Gilis, J.-L. *Attacks against the WiFi protocols WEP and WPA*. 2010.

[10] Fitzpatrick, J. *The Difference Between WEP, WPA, and WPA2 Wi-Fi Passwords* [online]. Srpen 2017 [cit. 2021-01-23]. Dostupné z: https://www.howtogeek.com/167783/htg-explains-the-difference-between-wep-wpa-and-wpa2-wireless-encryption-and-why-it-matters/.

[11] Gallagher, J. *PMKID Dumping: WiFi Password Attacks are Easier Than Previously Thought*. Srpen 2020 [cit. 2021-1-26]. Dostupné z: https://www.privateinternetaccess.com/blog/pmkid-dumping-wifi-password-attacks-are-easier-than-previously-thought/.

[12] Kumkar, V., Tiwari, A., Tiwari, P., Gupta, A. a Shrawne, S. Vulnerabilities of Wireless Security protocols (WEP and WPA2). *International Journal of Advanced Research in Computer Engineering & Technology*. Leden 2012, sv. 1.

[13] Lamers, E., Dijksman, R., Vegt, A. van der, Sarode, M. a Laat, C. de. Securing Home Wi-Fi with WPA3 Personal. In: *2021 IEEE 18th Annual Consumer Communications Networking Conference (CCNC)*. 2021, s. 1–8. DOI: 10.1109/CCNC49032.2021.9369629.

[14] Lashkari, A. H., Towhidi, F. a Hosseini, R. S. Wired Equivalent Privacy (WEP). In: *2009 International Conference on Future Computer and Communication*. 2009, s. 492–495. DOI: 10.1109/ICFCC.2009.32.

[15] Lounis, K. a Zulkernine, M. Bad-Token: Denial of Service Attacks on WPA3. In: *Proceedings of the 12th International Conference on Security of Information and Networks*. New York, NY, USA: Association for Computing Machinery, 2019. SIN '19. DOI: 10.1145/3357613.3357629. ISBN 9781450372428.

[16] Radivilova, T. a Hassan, H. A. Test for penetration in Wi-Fi network: Attacks on WPA2-PSK and WPA2-enterprise. In: *2017 International Conference on Information and Telecommunication Technologies and Radio Electronics (UkrMiCo)*. 2017, s. 1–4. DOI: 10.1109/UkrMiCo.2017.8095429.

[17] Raju, K. V. K., Vallikumari, V. a Raju, K. Modeling and analysis of IEEE 802.11i WPA-PSK authentication protocol. In: *2011 3rd International Conference on Electronics Computer Technology*. 2011, sv. 5, s. 72–76. DOI: 10.1109/ICECTECH.2011.5941959.

[18] Reddy, B. a Srikanth, V. Review on Wireless Security Protocols (WEP, WPA, WPA2 & WPA3). *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*. Červenec 2019, s. 28–35. DOI: 10.32628/CSEIT1953127.

[19] Sanatinia, A., Narain, S. a Noubir, G. Wireless spreading of WiFi APs infections using WPS flaws: An epidemiological and experimental study. In: *2013 IEEE Conference on Communications and Network Security (CNS)*. 2013, s. 430–437. DOI: 10.1109/CNS.2013.6682757.

[20] Tews, E. a Beck, M. Practical Attacks against WEP and WPA. In: *Proceedings of the Second ACM Conference on Wireless Network Security*. New York, NY, USA: Association for Computing Machinery, 2009, s. 79–86. WiSec '09. DOI: 10.1145/1514274.1514286. ISBN 9781605584607.

[21] Vibhuti, S. *IEEE 802.11 WEP (Wired Equivalent Privacy) concepts and vulnerability*. 2005 [cit. 2021-01-23]. Dostupné z: http://www.cs.sjsu.edu/faculty/stamp/CS265/projects/Spr05/papers/WEP.pdf.