Seminarska naloga KRV

Relay napad na DESFire EV1

Andrej Burja 63140024

Uvod

Relay napadi so čedalje bolj pogosti pri kraji denarja z brezstičnih bančnih kartic in pri kraji avtomobilov, ki vsebujejo dodatno funkcionalnost "*KeyLess*". Ker se podobna tehnologija uporablja tudi pri kontroli dostopa na fakulteti, je vprašljiva varnost kabinetov, laboratorijev in računalniških učilnic.

Ali je sistem za kontrolo dostopa na fakulteti ranljiv na relay napad? To je bilo glavno vprašanje, ki sem si ga zastavil pri realizaciji te seminarske naloge. Odgovor je DA, sistem je ranljiv.

Za izvedbo relay napada sta potrebna dva glavna člena, škodljiv čitalec (angl. *mole, ki* komunicira s ključkom) in posrednik (angl. *proxy, ki* se pretvarja, da je ključ). Ta glavna člena si med seboj izmenjujeta komunikacijo. Tako lahko napadalec prestreza, dodaja in spreminja vso komunikacijo med čitalcem in odzivnikom (ključem). Z drugimi besedami bi relay napad opisali kot napad, ki omogoča komunikacijo odzivnik – čitalec na daljši razdalji.

V tej seminarski nalogi bom izdelal/predelal dve napravi ter izvedel relay napad na fakulteti. Pri tem bom uporabil naprave: Proxmark3, čitalec PN532, mikrokontroler WeMos D1 mini in računalnik. Posnetek napada je dostopen na tej povezavi [1], koda pa je dostopna na GitHub povezavi [2].

Primer napada: Profesor sedi za mizo in pije kavo. Pridruži sem mu študentka Alica, ki ima v svoji torbici veliko anteno, priklopljeno na škodljiv čitalec. Ta lahko komunicira z odzivniki (ključem) na razdalji do 50 cm. Nadstropje višje pred zaklenjenim kabinetom profesorja čaka Bob, v roki ima napravo, ki simulira odzivnik. Ta naprava je preko Wi-Fi povezave povezana s škodljivim čitalcem, s katerim si izmenjujeta komunikacijo. Ko je torbica dovolj blizu ključu profesorja, škodljivi čitalec začne komunikacijo s ključem in napravo, ki jo ima Bob prislonjeno na čitalec pred kabinetom. Potrebna avtentikacija se izvede v sekundi. Posledično ima Bob sedaj dostop od odklenjenega kabineta.

Vemo, da osebje na fakulteti uporablja odzivnike DESFire EV1, ki so ranljivi na relay napad, vendar pa se ranljivosti da izogniti, če je čitalec nastavljiv in pravilno nastavljen. Da je DESFire EV1 ranljiv na relay napad, je znano že več let, zato je podjetje NXP, ki razvija to tehnologijo, že razvilo odzivnike nove generacije (EV2 in EV3), ki niso ranljivi. DESFire EV1 je na trgu že od novembra 2006, izboljšano verzijo EV2, pa so javnosti predstavili marca 2016. Pred DESFire EV1 so bili odzivniki DESFire (brez druge oznake), pri katerih je bilo možno pridobiti ključ na podlagi analize porabe odzivnika [3].

V seminarski nalogi bom opisal, kako delujejo odzivniki RFID, bolj podrobno bom opisal odzivnik tipa DESFire EV1 ter njegove pomembne lastnosti: »aplikacije«, datoteke, ključe itn. V poglavju *Postopek avtentikacije* bom po korakih opisal, avtentikacijo s ključem AES. Nadaljeval bom s poglavjem *Relay*, v katerem bom predstavil relay napad, ter kako delujejo orodja, ki so bila programsko spremenjena za izvedbo relay napada. Da bo postopek relay napada bolj razumljiv, bom nadaljeval s poglavjem *Lastnosti odzivnika, ki ga uporablja osebje na fakulteti*, v katerem bom predstavil programsko vsebino odzivnika. V poglavju *Komunikacija med čitalcem in odzivnikom* bom podrobno opisal (na fakulteti) zajeto komunikacijo med čitalcem in odzivnikom. Relay napad je odvisen od časa, zato je poglavje *Čas* namenjeno predvsem razumevanju, zakaj je relay napad sploh možen ter katere podatke je bilo treba

spremeniti pri komunikaciji, da je relay napad uspel. Na koncu poglavja se bom osredotočil na zaščito pred relay napadi. Seminarska naloga se konča z zaključkom in ugotovitvami.

RFID

RFID ali radiofrekvenčna identifikacija (angl. *Radio-frequency identification*) uporablja elektromagnetno polje za identifikacijo med odzivnikom (angl. *tag*) in čitalcem. Poznamo pasivne in aktivne odzivnike. Aktivni za delovanje uporabljajo baterijo, kar jim omogoča komunikacijo na daljše razdalje, pasivni pa ne vsebujejo baterije, ker se njihovo vezje napaja s pomočjo radijskih valov, ki jih oddaja bralna naprava (čitalec). Odzivniki morajo zato biti v bližini čitalca (5-10 cm), da lahko delujejo. Odzivnik je sestavljen iz dveh glavnih delov antene in mikrokontrolerja. Antena je po navadi večja od mikrokontrolerja, njena naloga pa ni samo, da se preko nje sprejema in oddaja radijske valove, ampak je namenjena tudi temu, da izkorišča radijske valove čitalca za napajanje mikrokontrolerja. Mikrokontroler v odzivniku vsebuje: oddajnik, sprejemnik, procesor, pomnilnik in druga orodja (glej sliko 1).

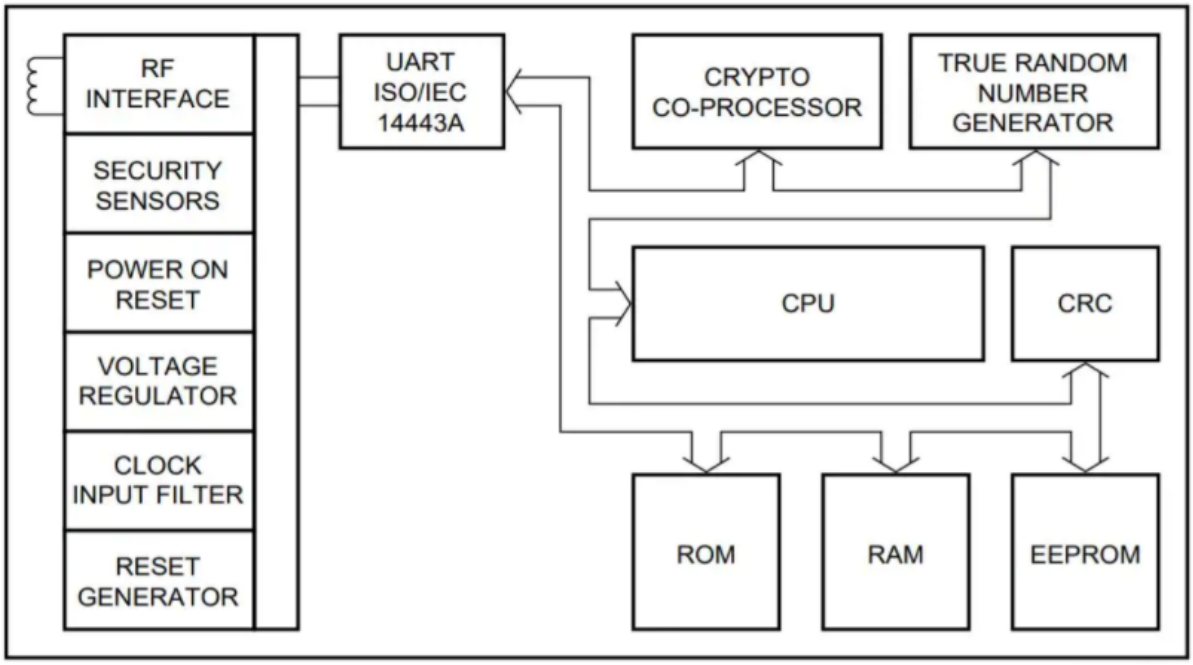
Pasivni odzivniki se uporabljajo za prepoznavanje artiklov v trgovini, v potniškem prometu, bančnih karticah, označevanje živali, sprejemanje pošiljk, pri izposoji knjig (tudi v knjižnici na fakulteti (FRI)), pri kontroli dostopa (do zaščitenih prostorov, parkirišč, garaž), v ključkih za avtomate itd. Odzivniki so lahko različnih oblik in velikosti. Največkrat jih najdemo v etiketah, karticah, obeskih itn. V prihodnosti pa naj bi nadomestili črtne kode.

MIFARE DESFire EV1

Odzivnik tipa DESFire je pasivni odzivnik, ki komunicira s čitalcem na frekvenci 13,56 MHz. Uporabljajo se v javnem prometu (LPP, slovenske železnice itn.), plačilnih karticah, pri kontrolah dostopa do zaščitenih prostorov, parkirišč, garaž itd.

DESFire je skladen z vsemi štirimi standardi ISO/IEC 14443A. Z odzivnikom lahko komuniciramo tudi preko ISO/IEC 7816-4 ukazov.

Odzivnik ima vgrajen sistem za medsebojno preverjanje pristnosti s tremi prehodi (angl. *three-pass authentication*). V mikrokontrolerju je pomnilnik, ki je lahko različnih velikosti (2 kB, 4 kB ali 8 kB), največkrat 4 kB. V pomnilnik EEPROM (*electrically erasable programmable read-only memory*) lahko shranimo do 28 različnih »aplikacij« in do 32 datotek na vsako »aplikacijo«. Aplikacija je spremenljiva struktura, ki je odgovorna za dostop do različnih datotek, preko katerih lahko uporabniki izvajajo različne storitve, kot so avtentikacija, plačevanje, shranjevanje podatkov itn. Tako je en odzivnik DESFire lahko uporabljen za več namenov (več različnih storitev). Vsaka aplikacija ima lahko 14 ključev, celoten odzivnik pa ima en glavni ključ. Kot pove že samo ime DESFire, se za varnost uporablja DES. Izbiramo lahko med DES (56 bit), 2K3DES (112 bit), 3K3DES (168 bit) in AES (128 bit) šifrirnimi standardi. Za avtentičnost podatkov se uporablja 8 bajtov velik CMAC. Mikrokontroler vsebuje tudi generator naključnih števil, ki predstavlja pomembno vlogo pri avtentikaciji. Vsak odzivnik ima unikatno enolično številko dolžine 7 bajtov, s katero se predstavi čitalcu. Na zadnji strani seminarske naloge je tudi slika, ki prikazuje vse informacije odzivnika, kakršne uporablja osebje na fakulteti [4] [5] [6].



Slika 1 Struktura mikrokrmilnika DESFire [6]

Aplikacije in ključi

Vsaka aplikacija na odzivniku DESFire ima 3 bajte dolg naslov, s katerim lahko naslovimo aplikacijo.

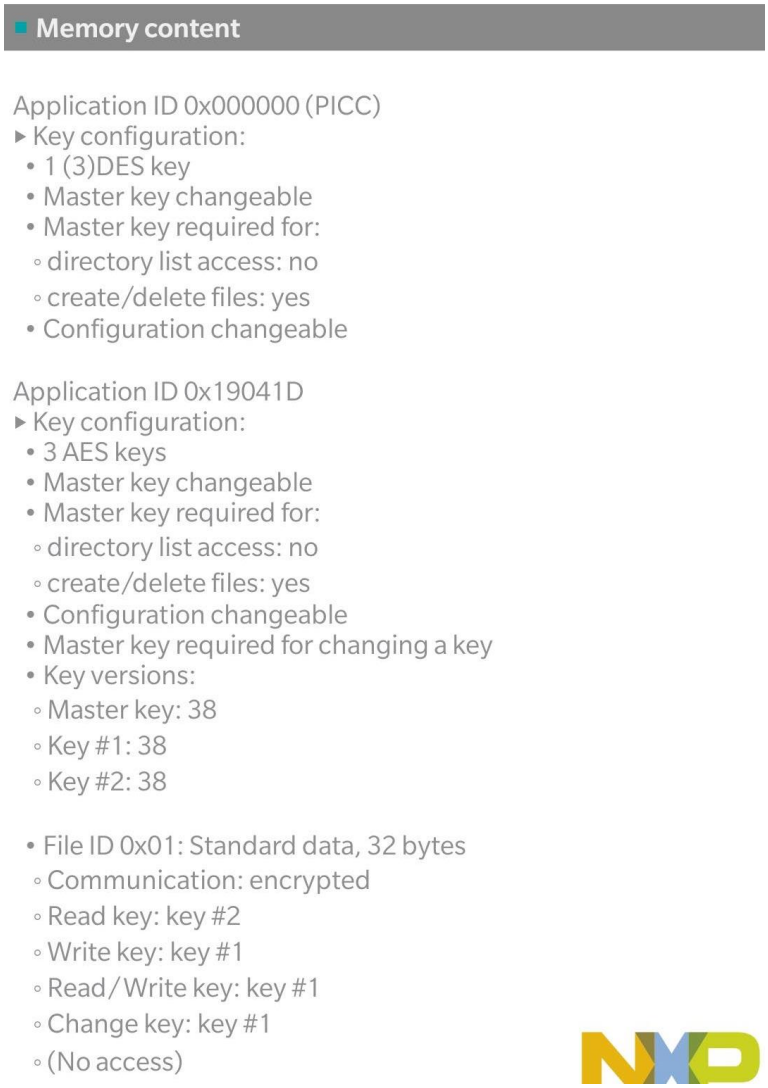
Aplikacija na naslovu 0x000000 je privzeto nameščena in vsebuje dostopne informacije na področju celotnega odzivnika. Vsebuje en ključ, s katerim lahko spreminjamo nastavitve o dostopu. Možno je celo nastaviti, da ključa ni možno spremeniti in da so vse trenutne nastavitve fiksne (slika 2 prikazuje podatke na odzivnikih, ki se uporabljajo na fakulteti).

Z glavnim ključem lahko nastavimo: ali je dovoljeno spreminjanje ključa, ali je glavni ključ potreben za ustvarjanje in brisanje aplikacij, ali za izpis seznama vseh aplikacij potrebujemo glavni ključ in ali lahko spreminjamo nastavitve.

Podobne nastavitve najdemo tudi na nivoju ene aplikacije, le da v tem primeru nastavitve veljajo za datoteke. Aplikacije imajo lahko tudi glavni ključ, ki je zadolžen za naslednje nastavitve: ali se ključi lahko spreminjajo, ali je za kreiranje in brisanje datotek potreben glavni ključ aplikacije, ali lahko brez ključa pridobimo seznam vseh datotek v aplikaciji in ali je možno te nastavitve spreminjati.

Kot rečeno, ima vsaka aplikacija lahko do 14 ključev, ti ključi so namenjeni datotekam.

V vsaki datoteki lahko v varnostnih nastavitvah določimo: kateri ključ lahko bere datoteko, kateri ključ lahko piše v datoteko, kateri ključ lahko bere in piše v datoteko in kateri ključ lahko spremeni nastavitve datoteke. Pri vseh teh nastavitvah je možna tudi izbira »free access«, ki odobri dostop do datoteke brez ključa [7] [8] [5] [4].



Slika 2 Podatki o aplikacijah, ključih in datotekah na odzivniku, ki ga uporablja osebje na fakulteti.

Postopek avtentikacije

Medsebojno preverjanje pristnosti s tremi prehodi (angl. *three-pass authentication*) z uporabo AES deluje v naslednjih korakih:

1. Čitalec posreduje ukaz za avtentikacijo z določenim ključem.
2. Odzivnik generira 16 bajtov veliko naključno število B in ga šifrira z izbranim AES ključem. Rezultat je nato posredovan čitalcu.
3. Čitalec prejme 16 bajtov velik odgovor in ga dešifrira s ključem, ki je na strani čitalca. Tako dobi naključno število B. Pri tem je IV (*Initialization vector*) na vseh bajtih nastavljen na 00. Čitalec nato generira svoje naključno število A, pridobljeno število B pa zamakne za en bajt v levo. A in spremenjeni B nato združi skupaj v 32 bajtov, ki jih šifrira z AES ključem, rezultat pa pošlje odzivniku. IV je pri šifriranju enak šifriranemu B številu, ki ga je pred tem prejel od odzivnika.
4. Odzivnik prejme 32 bajtov in jih dešifrira. Odzivnik nato preveri, ali je zadnjih 16 bajtov enakih zamaknjenemu številu B. Če niso enaki, je bila avtentikacija neuspešna in odzivnik čitalcu posreduje napako. Če pa je število enako, odzivnik ve, da ima čitalec pravi ključ. Nadaljuje

tako, da prvih 16 bajtov števila A zamakne za 1 bajt v levo, rezultat šifrira in ga posreduje čitalcu.

5. Čitalec prejme 16 bajtov, ki jih dešifrira, IV pa je v tem primeru nastavljen na zadnjih 16 bajtov, ki jih je čitalec poslal odzivniku. Čitalec nato preveri, ali je zamaknjeno število A enako prejetemu dešifriranemu številu. Neenakost pomeni, da je bila avtentikacija neuspešna. Če pa sta števili enaki, čitalec ve, da imata oba isti ključ.

Na koncu obe strani izračunata ključ za nadaljnjo komunikacijo. Sejni ključ je enak prvim 4 bajtom A + prvim 4 bajtom B + zadnjim 4 bajtom A + zadnjim 4 bajtom B [5] [4].

Po avtentikaciji je lahko komunikacija šifrirana ali pa tudi ne. Vse je odvisno od nastavitev dostopa. V primeru, da so ukazi šifrirani, se uporablja CMAC [4]. Ker se IV posodablja, je zagotovljeno, da MITM (*man in the middle*) in *replay* napadov ni možno izvesti.

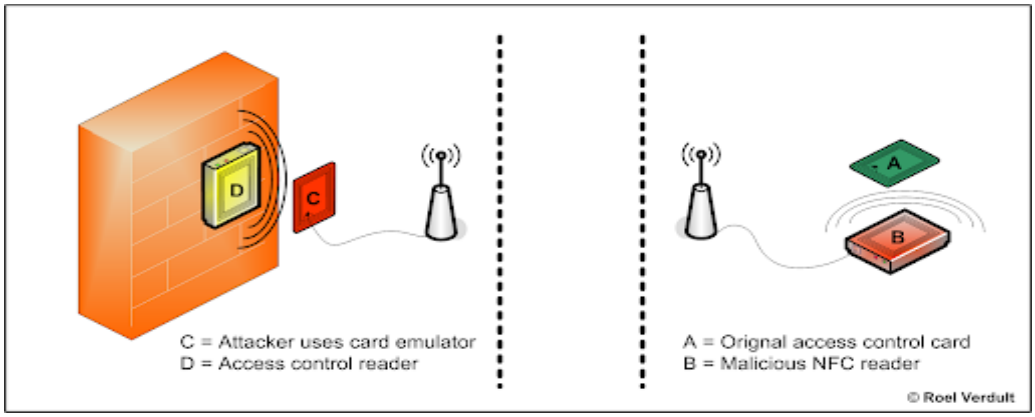
Relay

Za izvedbo relay napada sta potrebna dva glavna člena, škodljiv čitalec (*mole*) in posrednik (*proxy*).

Škodljiv čitalec je naprava, ki komunicira z odzivnikom. Ukaze prejema s strani posrednika in jih izvede nad odzivnikom. Odgovore odzivnika nato posreduje nazaj posredniški napravi.

Posrednik je naprava, ki komunicira s čitalcem, pri čemer se pretvarja, da je pravi odzivnik. Vse ukaze, ki jih čitalec posreduje, preusmeri škodljivemu čitalcu, ta pa mu odgovori z rezultati odzivnika.

Pri mojem prototipu je posredniška naprava odvisna od samega računalnika, zato je bilo treba izdelati program, ki izmenjuje ukaze in odgovore med posredniško napravo in škodljivim čitalcem. Na ta način lahko tudi lažje izvedemo MITM.

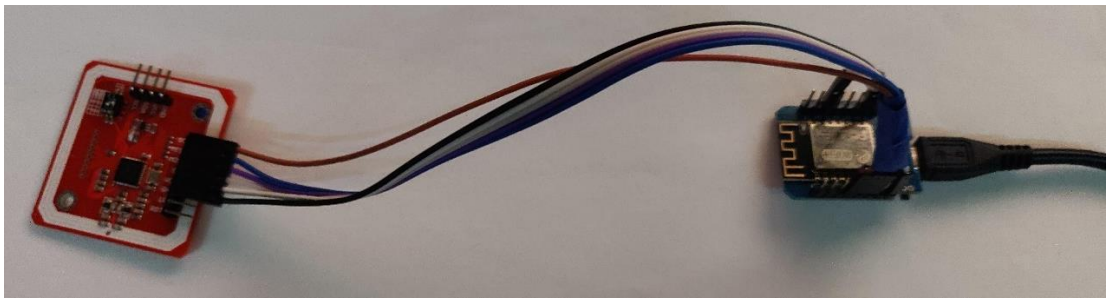


Slika 3 Relay napad [9]

Za izvedbo relay napada so bila izdelana, spremenjena in napisana naslednja orodja:

- Škodljiv čitalec. Je naprava, ki se uporablja za neposredno komunikacijo z odzivnikom. Ta naprava je sestavljena iz dveh glavnih komponent: mikrokontrolerja *WeMos D1 mini* in čitalca *PN532* (na sliki 4). Komponenti sta med seboj povezani preko SPI (*Serial Peripheral Interface Bus*) protokola. Mikrokontroler pa je povezan tudi z računalnikom preko serijskega vmesnika. Naprava na začetku prebere UID (enolična identifikacijska številka) odzivnika in njegove nastavitve in jih posreduje računalniku, na katerem je pognan program, ki skrbi za prenos podatkov med škodljivim čitalcem in posredniško napravo. Pri programiranju

mikrokontrolerja *WeMos D1 mini* je bilo uporabljeno programsko orodje *Arduino*. Za pomoč pri komunikaciji mikrokontroler - čitalec pa je bila uporabljena knjižnica *PN532* [10].



Slika 4 Škodljivi čitalec (levo čitalec PN532, desno mikrokontroler WeMos D1 Mini)

- Program na računalniku. Spisan je v programskem jeziku C# in skrbi za medsebojno komunikacijo med škodljivim čitalcem in posredniško napravo. Program je preko serijskega vmesnika povezan s škodljivim čitalcem, s posredniško napravo pa preko proxmark3 *client* konzole. Uporabnik lahko preko konzolne aplikacije na računalniku spremlja vso komunikacijo, ki se izvaja.
- Posredniška naprava. Za izvedbo posredniške naprave je bila uporabljena naprava Proxmark3 (na sliki 5). Ta naprava je na trgu prisotna že več let, uporabljajo jo kot univerzalno orodje na področju tehnologije RFID. S to napravo lahko zajemamo komunikacijo med pravim čitalcem in pravim odzivnikom, beremo odzivnike itn. Naprava Proxmark je v celoti odprtokodna, tako da uporabniki lahko po potrebi kodo za svoje namene spremenijo. Omogoča nam tudi simulacijo odzivnika, zaradi česar sem jo tudi uporabil pri tej seminarski nalogi. Da je naprava prevzela vlogo posredniške naprave, je bilo potrebno predelati programje, ki je gnano na mikrokontrolerju AT91SAM7S512 in programje, ki je gnano s strani računalnika, ta pa skrbi za komunikacijo z napravo Proxmark3 [11].

Slika 5 Proxmark3 Easy

Lastnosti odzivnika, ki ga uporablja osebje na fakulteti

Osebje na fakulteti uporablja odzivnike tipa DESFire EV1, ki jih je razvilo podjetje NXP. Odzivnik ima nastavljen en glavni ključ (3DES), ki se uporablja pri spreminjanju ključa, kreiranju novih aplikacij in spreminjanju konfiguracij. Da pridemo do seznama vseh aplikacij, pa je nastavljen *free access*, tako da za branje avtentikacije ne potrebujemo. Odzivniki imajo 4 kB prostora na njih pa je nameščena ena aplikacija, ki se uporablja pri avtentikaciji. Ta aplikacija ima nameščene tri ključe (AES), pri avtentikaciji pa uporablja ključ št. 1 (ključ št. 0 je glavni ključ in je namenjen spreminjanju nastavitev aplikacije). Naslov aplikacije je 0x19041D, v njej pa je shranjena ena datoteka, velika 32 bajtov.

Datoteka z naslovom 0x01 ima omogoča naslednje nastavitve: pri prenosu je povezava šifrirana, ključ št. 2 lahko le bere, kar je v datoteki, ključ št. 1 pa lahko bere, piše in spreminja nastavitve (pravice dostopa) datoteke. Ti podatki so prikazani na sliki 2.

Komunikacija med čitalcem in odzivnikom (pridobljena na fakulteti)

Komunikacija je bila zajeta na fakulteti s pomočjo Proxmark3 orodja z ukazom »*hf 14a snoop*« [11]. Ukaz lahko bere vso komunikacijo, ki deluje po protokolu ISO 14443A. Spodnja struktura komunikacije (slika 6) je enaka tako pri zaklepanju kot odklepanju vrat. Struktura se ne spremeni niti ko oseba poskuša odpreti vrata, do katerih nima dostopa.

1. Čitalec neprestano pošilja 7-biten ukaz »*26*« ter čaka na odgovor odzivnika.
2. Odzivnik odgovori »*44 03*« kar pomeni, da je odzivnik tipa DESFire.
3. Nato se začne postopek preprečevanja trkov (*anticollision*), kar omogoča, da lahko čitalec govori z več odzivniki v svojem dosegu. Odzivnik najprej posreduje prve 3 bajte svojega enoličnega naslova. Čitalec izbere vse odzivnike, ki vsebujejo ta naslov. Nato pa od teh zahteva še preostali naslov (zadnje 4 bajte UID), ki ga izbere v naslednjem koraku. Vsi neizbrani odzivniki ostanejo tiho, izbrani pa lahko odgovarja na naslednje ukaze.
4. V naslednjem ukazu čitalec pošlje ukaz RATS (*Request for Answer To Select*), ki zahteva ATS (*Answer To Select*). Gre za parametre, s katerimi odzivnik čitalcu poda pogoje, pod kakršnimi lahko komunicira z njim. Vmes je tudi polovica bajta namenjena FWT (*Frame Waiting Time*), ki igra pomembno vlogo pri izvajanju relay napada (več o tem v naslednjem poglavju *Čas*).
5. Čitalec izbere (0x5A) aplikacijo 0x19041D, odzivnik pa izbiro potrdi (0x00).
6. Z naslednjim ukazom čitalec začne postopek avtentikacije (0xAA -> AES avtentikacija) s ključem št. 1.
7. Odzivnik nato posreduje šifrirano 16 bajtov veliko naključno število (B). Ker odzivnik ni sposoben posredovati vseh podatkov v enem kosu, pošlje podatke večkrat (v prvem paketu jih posreduje 12 v drugem pa 4, pri vsakem paketu sta zadnja 2 bajta namenjena kontroli prenosa - CRC).
8. Čitalci imajo bolj zmogljive procesorje, zato lahko v enem samem paketu posredujejo vse bajte. V našem primeru je čitalec posredoval 36 bajtov, od katerih je srednjih 32 bajtov šifriranih podatkov. Vanje je zapisano naključno število A (ki ga je generiral čitalec) in naključno število B, ki je zamaknjeno en bajt v levo.
9. Odzivnik dešifrira in preveri, ali imata s čitalcem isti ključ. Nato čitalcu posreduje šifrirano naključno število A, ki je zamaknjeno za 1 bajt.
10. Na koncu obe strani nastavita sejni ključ, preko katerega poteka naslednja šifrirana komunikacija. Ukazi so v tem primeru nešifrirani, odgovori odzivnika pa šifrirani.
11. Čitalec posreduje ukaz za branje datoteke (0xBD), pri tem izbere datoteko 0x01, zamik je nastavljen na 0, dolžina pa je zapisana pod 0x11, kar je desetiško 17. V dokumentaciji je navedeno, da če je zamik (*offset*) nastavljen na 0, potem se prebere celotna datoteka (32 bajtov), ki se prenese v treh ločenih paketih.
12. Čitalec z ukazom 0xC2 prekine komunikacijo (angl. *deselect*), odzivnik pa mu odvrne z enakim ukazom.
13. Naslednji ukazi so namenjeni le temu, da čitalec preverja ali je odzivnik še vedno prisoten v njegovem območju [8] [4].

Vsa ta komunikacija se izvede v 150 ms. Pri skoraj vseh ukazih (razen na začetku komunikacije) se uporablja CRC za kontrolo prenosa, čemur sta namenjena zadnja 2 bajta v paketu.

[2]Slika 6 Komunikacija med čitalcem in odzivnikom (komunikacija je bila zajeta na fakulteti)

Čas

Največja težava pri izvajanju relay napada je čas. Če imamo časa dovolj, bo relay napad uspel, v nasprotnem primeru pa napad ne bo možen. Pri izvajanju relay napada lahko uporabimo dve vrsti zakasnitve.

FWT

Frame Waiting Time, v nekaterih dokumentacijah tudi FWI. Gre za podatek, ki je del parametrov ATS. FWT pove čitalcu, da bo odgovor posredovan v določenem času (maksimalno čakanje na odgovor odzivnika). FWT podatek je velik pol bajta, v našem primeru je nastavljen na 0x8, kar pomeni približno 77 ms. Nastavljen je lahko od 0 pa tja do 14 [5] [8]. Izračuna se po naslednji enačbi:

$$256 * \left(\frac{16}{frekvenca}\right) * 2^{FWT}$$

Primer za FWT 14, ki je tudi maksimalna dolžina čakanja:

$$256 * \left(\frac{16}{13.56 * 10^6}\right) * 2^{14} = 4.949 sekund$$

Maksimalna dolžina čakanja na odgovor odzivnika je 4.949 sekund [8]. Pri izdelavi posredniške naprave je v kodi spisano, da se originalna vrednost (0x8) prepiše z novo večjo vrednostjo (0xE = 4,9 sekunde).

WTX (*Frame Waiting Time Extension*)

Če odzivnik potrebuje več časa, kot je nastavljena vrednost FWT, lahko posreduje čitalcu paket, ki ukaže, naj počaka. Čitalec na ta paket odgovori z istim paketom. V paketu velikosti enega bajta je treba nastaviti prvih 6 bitov ukaza, v katerega zapišemo število, ki bo pomnoženo s trenutno nastavljeno vrednostjo FWT. Če je začasen FWT večji od maksimalne vrednosti FWT, se začasni FWT nastavi na maksimalno vrednost FWT (4,9 sekunde) [8].

Zaščita pred relay napadi

Zaščitimo se lahko tako, da uporabljamo novejše različice DESFire EV2 in EV3. Obe različici vsebujeta tehnologijo preverjanja bližine, ki naj bi ščitila pred relay napadi [7].

Drugi način zaščite je odvisen od čitalca. Čitalec mora biti nastavljiv tako, da lahko omejimo čas čakanja na odgovor, tako ignoriramo FWT vrednost in pakete WTX. To pa je v nasprotju s standardom. Po mojih ugotovitvah so vsi ukazi izvedeni v manj kot 10 milisekundah (odgovori odzivnika v mojem primeru). Tako nizka vrednost bi bila dovolj, da relay napad skoraj ne bi bil več mogoč, še posebej, če posredniška naprava in škodljivi čitalec uporabljata druge oblike brezžične komunikacije.

Uporabniki ranljivih odzivnikov se lahko pred relay napad zaščitijo tako, da uporabljajo denarnice, vrečke ali torbice iz posebnih materialov, ki motijo komunikacijo.

Zaključek

Cilj seminarske naloge je bil dosežen. Relay napad je bil uspešno izveden na fakulteti. Programje za izvedbo relay napada je bilo izdelano na način, da lahko napad izvedemo večkrat zapored. Ko je odzivnik v območju škodljivega čitalca in posredniška naprava v območju terminala za odpiranje vrat, se relay napad izvede v približno 5 sekundah (ta čas bi bilo možno zmanjšati). Prava komunikacija med čitalcem in odzivnikom je izvedena v približno 150 milisekundah, relay napad pa potrebuje malo manj kot sekundo. Tudi ta čas bi bilo možno skrajšati, če bi se kodi res posvetili. Največ od tega časa vzame *proxmark client,* preko katerega se komunicira z napravo. Za hitrejšo komunikacijo bi bilo treba uporabiti povezavo UART. V primeru, da bi imeli originalno različico proxmark, bi lahko uporabili tudi SPI povezavo. Novejše različice proxmark že vsebujejo *Bluetooth*.

Tudi orodja, ki sem jih uporabil pri napadu bi lahko enostavno predelali, da bi delovala brezžično. Najbolj enostaven način bi bil, da bi uporabili *WiFi* modul na mikrokontrolerju *WeMos* (že vgrajen). Na drugi strani pa bi uporabili *WiFi* na računalniku, podatke bi si izmenjevali s pomočjo UDP paketov.

Med testiranjem sem izvedel tudi manjši napad MITM, pri katerem sem spremenil originalni UID odzivnika. S tem sem ugotovil, da v tem primeru 7 bajtov velik ID odzivnika ni pomemben pri avtentikaciji. Najbolj verjetno je torej, da imajo vsi odzivniki, ki jih uporablja osebje na fakulteti enake AES ključe. Kar je tukaj unikatno, je verjetno 32 bajtov velika datoteka. Vsebina te datoteka pa nam je neznana, saj je komunikacija med prenosom te datoteke šifrirana. Torej je prava avtentikacija narejena preko te datoteke.

Literatura

[1] A. Burja, „YouTube,“ [Elektronski]. Available: https://www.youtube.com/watch?v=IediWHW2NhA ALI tiny.cc/desfire.

[2] A. Burja, „GitHub,“ [Elektronski]. Available: https://github.com/burja8x/RelayDESFireEV1.

[3] C. P. -. R.-U. B. David Oswald, „Iacr,“ 2011. [Elektronski]. Available: https://www.iacr.org/workshops/ches/ches2011/presentations/Session%205/CHES2011_Session5_1.pdf. [Poskus dostopa 9 9 2020].

[4] TheRealRevK, „NXP MIFARE DESFire Protocol“.

[5] PHILIPS, „DATA SHEET mifare DESFire,“ april 2004. [Elektronski]. Available: http://neteril.org/files/M075031_desfire.pdf . [Poskus dostopa 24 8 2020].

[6] NXP, „MIFARE® DESFire® EV1,“ [Elektronski]. Available: https://www.nxp.com/products/rfid-nfc/mifare-hf/mifare-desfire/mifare-desfire-ev1:MIFARE_DESFIRE_EV1_2K_8K. [Poskus dostopa 24 8 2020].

[7] NXP, „MIFARE DESFire EV2,“ 2019. [Elektronski]. Available: https://www.nxp.com/products/rfid-nfc/mifare-hf/mifare-desfire/mifare-desfire-ev2:MIFARE_DESFIRE_EV2_2K_8K. [Poskus dostopa 24 8 2020].

[8] ISO, „INTERNATIONAL STANDARD ISO/IEC 14443-4,“ julij 2018. [Elektronski]. Available: https://sgp1.digitaloceanspaces.com/proletarian-library/books/cd1f6cf708a2c4efdca72d15e1e65337.pdf. [Poskus dostopa 24 8 2020].

[9] nfc-tools, „Relay attack access control,“ [Elektronski]. Available: http://nfc-tools.org/index.php/File:Relay_attack_access_control.png. [Poskus dostopa 24 8 2020].

[10] Seeed-Studio, „GitHub,“ [Elektronski]. Available: https://github.com/Seeed-Studio/PN532. [Poskus dostopa 24 8 2020].

[11] „GitHub - Proxmark3 Iceman,“ [Elektronski]. Available: https://github.com/RfidResearchGroup/proxmark3. [Poskus dostopa 24 8 2020].

Memory information

Size: 4 kB
Available: 4.5 kB

IC detailed information

Capacitance: 17 pF

Version information

Vendor ID: NXP
Hardware info:
- Type/subtype: 0x01/0x01
- Version: 1.0
- Storage size: 4096 bytes
- Protocol: ISO/IEC 14443-2 and -3

Software info:
- Type/subtype: 0x01/0x01
- Version: 1.4
- Storage size: 4096 bytes
- Protocol: ISO/IEC 14443-3 and -4

Batch no: 0xBA74579ED0
Production date: week 17, 2016

Technologies supported

ISO/IEC 7816-4 compatible
Native DESFire APDU framing
ISO/IEC 14443-4 (Type A) compatible
ISO/IEC 14443-3 (Type A) compatible
ISO/IEC 14443-2 (Type A) compatible

Detailed protocol information

ID: [redacted]
ATQA: 0x4403
SAK: 0x20
ATS: 0x06757781028000
- Max. accepted frame size: 64 bytes (FSCI: 5)
- Supported receive rates:
  - 106, 212, 424, 848 kbit/s (DR: 1, 2, 4, 8)
- Supported send rates:
  - 106, 212, 424, 848 kbit/s (DS: 1, 2, 4, 8)
- Different send and receive rates supported
- SFGT: 604.1 µs (SFGI: 1)
- FWT: 77.33 ms (FWI: 8)
- NAD not supported
- CID supported
- Historical bytes: 0x80 |·|

Slika 7 Podrobni podatki o odzivniku katerega uporablja osebje na fakulteti