ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

Dipartimento di Informatica — Scienza e Ingegneria
Corso di Laurea Triennale in Informatica

Validazione dei beacon frames per il rilevamento di attacchi Evil Twin: progettazione e implementazione di un sistema basato su nonce

Tesi di Laurea in Sicurezza Informatica

Relatore:
Prof. Marco Prandini

Presentata da:
Samuele Zucchini

Sessione Dicembre 2025
Anno Accademico 2024/2025

Indice

Capitolo 1

Introduzione

1.1 Motivazione del lavoro e risultati attesi

A partire dall'intento di approfondire la mia conoscenza delle reti WiFi e di comprendere al meglio i meccanismi che ne regolano il funzionamento, specialmente riguardo i processi di autenticazione tra client ed access point, ho ideato, progettato e sviluppato un sistema alternativo per il rilevamento di attacchi evil twin, il quale costituisce il punto focale della tesi. L'intento del lavoro svolto è dunque quello di acquisire dimestichezza con il processo di ricerca e sviluppo di un sistema fine alla protezione delle reti WiFi, illustrando la progettazione e l'implementazione di un'architettura che offra un approccio alternativo al rilevamento dei suddetti attacchi. L'esito ricercato dal progetto è un sistema funzionante che risponda positivamente alle specifiche introdotte, ma soprattutto una valutazione rilevante di vantaggi e svantaggi offerti dal sistema in questione, accompagnata dall'esplorazione di possibili sviluppi futuri e potenziali integrazioni. La collocazione della soluzione proposta rispetto ai prodotti in commercio che già prevengono questi tipi di attacchi è parte fondamentale dell'elaborato: ciò mira a mantenere chiara la direzione presa nell'arco dell'intero processo, delineando come e per quale ragione siano state preferite determinate soluzioni a discapito di altre.

L'organizzazione dell'elaborato prevede l'illustrazione della natura dell'attacco in questione allo stato attuale (ossia come esso venga perpetrato e come le soluzioni già esistenti ne contrastino la riuscita), della struttura ed il funzionamento della soluzione proposta, la documentazione dei test condotti e, infine, una serie di considerazioni finali circa le potenzialità e le criticità riscontrate nel percorso. Sono incluse, come anticipato, constatazioni su potenziali estensioni del progetto trattato ad ambiti e scenari analoghi a quelli trattati. Al fine di raggiungere risultati consistenti e poter trarre conclusioni rilevanti nell'ambito della sicurezza delle reti, ho anteposto alla fase di strutturazione del mio progetto una considerevole attività di studio del tema, approfondendo i concetti alla base dell'intera struttura realizzata.

Capitolo 2

Modello della minaccia

2.1 Introduzione sulla tecnologia WiFi

Per contestualizzare al meglio il lavoro svolto è opportuno fornire un'introduzione alla tecnologia WiFi, focalizzandoci sulle aree più cruciali per il sistema in oggetto. Le comunicazioni fondate su questo protocollo, intrattenute tra dispositivi che distinguiamo tra access point (AP) e station, ossia i client che si avvalgono dei servizi dei primi, seguono dinamiche e meccanismi definiti dallo standard IEEE 802.11, nel quale è descritta la natura della tecnologia WiFi, a partire dal livello fisico. Partendo proprio da questo punto, approfondiamo come i dispositivi coinvolti veicolino i propri scambi. Client e AP si avvalgono di opportune interfacce di rete e antenne per trasmettere onde radio, ristrette ad uno spettro di frequenze designato e suddiviso in canali, convertendo in segnali fisici i pacchetti che costituiscono le comunicazioni. Sono previste meccaniche che prevengono e gestiscono fenomeni come collisioni, deterioramento della qualità del segnale e spostamento dei client, fattori che non approfondiremo ulteriormente.

I pacchetti che concretizzano gli scambi tra i dispositivi possono essere distinti in tre categorie. La prima, quella dei data frames, costituisce il traffico dati IP vero e proprio, mentre la seconda, quella dei control frames, implementa lo scambio di messaggi a supporto dell'architettura stessa, consentendo una gestione ottimale delle trasmissioni e garantendo un funzionamento ideale dell'intero sistema. In quest'ultima categoria risaltano gli acknowledgements, frame che fungono da conferma di avvenuta ricezione di altri pacchetti, e RTS e CTS, rispettivamente requst-to-send e clear-to-send, segnali utilizzati dai dispositivi per coordinare la trasmissione, al fine di prevenire e ridurre le collisioni menzionate in precedenza. L'ultima categoria riguarda i management frames, pacchetti che supportano lo stabilimento e la gestione delle connessioni. Tra essi vi è il beacon, frame di estrema rilevanza per il sistema descritto dall'elaborato e le considerazioni da esso tratte, che serve lo scopo di informare le station della presenza nell'area della rete che lo trasmette. Esso contiene un considerevole quantitativo di informazioni, tra cui le sicurezza adottata dal network e la suite di algoritmi crittografici supportati, elementi fondamentali per consentire ai client di connettersi agli access point con successo.

Per trattare altri management frames, vediamo ora come viene instaurato il rapporto di autenticazione e associazione di un client ad una rete. I client che registrano beacon provenienti da reti in prossimità (o che sono consapevoli della loro presenza, in caso di

"hidden networks") intenzionati a stabilire una connessione con esse devono, secondo lo standard, seguire il processo di autenticazione e associazione: il primo consiste nell'invio, da parte della station, di una probe request, alla quale l'access point risponde con una probe response, passaggi che fungono solamente da riconoscimento della reciproca presenza ed operatività. Nella successiva fase di associazione vengono invece negoziati parametri relativi alla natura della connessione e della sua realizzazione, tra cui il livello di sicurezza implementato, le velocità di trasmissione supportate, eventuale protezione di management frames e altri parametri. In reti non protette, le station sono ora abilitate alla trasmissione e ricezione di traffico IP tramite la rete, cosa che invece non è ancora possibile in network che prevedono l'impiego di protocolli di sicurezza come WEP (Wired Equivalent Privacy), WPA (WiFi Protected Access) o versioni successive. In quei casi, un ulteriore layer di protezione è infatti implementato, richiedendo ai client di autenticarsi tramite le modalità previste dal protocollo designato. WPA2-Personal incarna la configurazione PSK (Pre-Shared-Key), ad oggi ancora fortemente diffuso seppur in esso siano state identificate molteplici vulnerabilità, prevede lo scambio di quattro messaggi che consente all'AP di verificare che la station sia effettivamente in possesso della chiave originale. Questo processo prende il nome di four-way-handshake, il quale consiste nella trasmissione di chiavi derivate, non esponendo dati sensibili in chiaro, scambio che può essere tuttavia catturato da attaccanti per condurre attacchi offline di bruteforce, tramite i quali è possibile risalire alla PSK. Una volta stabilito lo stato di associazione tra le due parti, è possibile indurre un client a dissociarsi o deautenticarsi dalla rete mediante disassociation e deauthentication frames, pacchetti coinvolti nella perpetrazione di attacchi evil twin, processo analizzato in seguito. L'autenticazione, nel caso di WPA-Enterprise, segue il medesimo flusso, ma si differenzia per la filosofia adottata in termini di sicurezza, essa è infatti incentrata su certificati che vengono gestiti, lato access point, da un server RADIUS (Remote Authentication Dial-In User Service) che si occupa di verificarne la validità. Probe request/response, deauth, disassociation, 4-way-handshake e fenomeni simili sono tutti associati a rispettivi pacchetti che rientrano nella categoria dei management frames.

2.2 Evil twin attacks

2.2.1 Funzionamento

Gli attacchi evil twin consistono nella creazione di un access point avente lo stesso SSID (Service Set Identifier) - e possibilmente lo stesso BSSID (Basic Service Set Identifier) tramite spoofing - di quello della vittima: l'attaccante si pone in prossimità del target e, mediante una serie di strumenti hardware (es. network adapter, pineapple) e software (es. airbase-ng, airgeddon), raccoglie informazioni su di esso, cattura un 4-way-handshake per poter instaurare connessioni spacciandosi per l'access point legittimo, crea una sua copia malevola che controlla ed infine induce i client (spesso mediante attacchi di deauthentication) a disconnettersi dall'AP originale per connettersi ad essa. La potenza del segnale rilevata dai dispositivi che cercano di connettersi ad una rete gioca un ruolo fondamentale per la riuscita dell'attacco, infatti, tra molteplici reti aventi lo stesso SSID e le medesime configurazioni di sicurezza, le probabilità che una station si connetta a quella avente un segnale più forte sono più elevate [**1**]. I dispositivi che intrattengono connessioni basate su WPA2-PSK non dispongono di alcun metodo per certificare l'identità dell'AP

con cui si interfacciano, dunque il solo rilevamento di una rete conosciuta (avente SSID relativo a connessioni stabilite in passato) è sufficiente a indurre un tentativo di associazione; questo è il motivo per cui questi attacchi sono possibili. Queste vulnerabilità sono ampiamente ridotte da WPA3, WPA-Enterprise e da sistemi analizzati in seguito, ma è doveroso specificare che WPA2 in modalità PSK, come anche dimostrato da alcune indagini, è ad oggi ancora fortemente diffuso [**2**] sia in realtà domestiche che aziendali, rendendo l'attacco in oggetto tutt'altro che obsoleto.

2.2.2 Ripercussioni sulla sicurezza dei client

L'intento dell'attaccante è quindi quello di far connettere i client alla propria macchina, ciò offre una serie di ulteriori attacchi che mettono a rischio i dati e la riservatezza degli utenti colpiti. La versione più basilare di questa minaccia è la creazione di un fake captive portal, ovvero l'indirizzamento degli utenti ad una pagina che richieda l'inserimento di username, password, indirizzi email ed altri dati sensibili, per esempio tentando di convincere la vittima che il passaggio sia necessario al fine di ottenere l'accesso alla rete. Una situazione del genere può essere sfruttata per ottenere sia password dell'AP vittima, "richiedendola" direttamente agli utenti, ma anche per scopi potenzialmente molto più dannosi, come l'ottenimento di dati bancari, per esempio chiedendo il pagamento di una quota per l'utilizzo della rete. Risulta tutt'altro che complesso clonare layout/pagine web di banche e piattaforme di pagamento al fine di ingannare gli utenti, inducendoli a credere di star compiendo una transazione sicura. Per questa variante che fa ampiamente uso di ingegneria sociale, non è necessario che l'attaccante disponga di un four-way-handshake relativo all'access point target, infatti è sufficiente attribuire un SSID che induca gli utenti a scegliere di collegarsi al network falso, potenzialmente anche aperto e privo di autenticazione WPA.

Un secondo scenario, altrettanto plausibile seppur maggiormente ostacolato dalle tecnologie oggi diffuse, prevede che l'attaccante, qui in possesso del four-way-handshake, impersoni "strettamente" la rete della vittima, rispondendo alle richieste di associazione dei client e osservando il traffico da essi generato. Il primo potenziale pericolo è indotto da connessioni non cifrate, infatti il gateway malevolo può ispezionare i contenuti trasmessi in chiaro tramite HTTP (Hypertext Transfer Protocol), minaccia altamente mitigata dall'imposizione di connessioni HTTPS, le quali possono comunque essere monitorate per tracciare il comportamento e gli indirizzi richiesti dalle vittime, situazione decisamente meno critica ma che espone comunque gli utenti a potenziali violazioni della privacy.

Un'ulteriore escalation dell'evil twin può essere identificata nell'attuazione di DNS spoofing [**3**], ossia della manipolazione delle richieste DNS (Domain Name System) inoltrate dalla vittima per condurla a pagine gestite dall'attaccante: un utente che intende visitare un sito legato a potenziali dati sensibili, inserendo richiedendo l'URL di un sito legittimo, può essere reindirizzato a domini fini a sottrarre credenziali. Tattica nota agli attaccanti è la creazione di domini aventi nomi che mimano o richiamano fortemente siti affidabili e ricollegabili a dati sensibili. Il primo caso di questa pratica, ad oggi fortemente mitigato, riguarda gli attacchi omografici: sfruttando caratteri provenienti da alfabeti differenti, quindi aventi codifiche UNICODE diverse, è infatti possibile creare indirizzi

che visivamente siano identici. Per esempio, dato un indirizzo come "www.banca.com", è possibile registrare un dominio a fini di phishing che risulti identico ad esso, nel quale le "a" codificate in ASCII presenti nella parola "banca" siano sostituite da "a" dell'alfabeto cirillico, simboli graficamente uguali ma codificati con identificativi diversi in UNICODE. Ad oggi i browser, i registri di dominio e sistemi di sicurezza mitigano altamente questo attacco adottando opportuni accorgimenti, mettendo in risalto i caratteri anomali mostrandone direttamente la codifica o notificando gli utenti di potenziali minacce, tuttavia esistono attacchi simili che bypassano queste contromisure.

Se l'attacco appena analizzato si fondava su un funzionamento non ottimale di browser e registri di dominio, il seguente fa invece leva sulla vulnerabilità degli utenti stessi. Nel contesto web, gli attacchi conosciuti come "Typosquatting" sono un tipo di minaccia che si fonda sulla creazione di domini (sempre a scopo di phishing) che richiamino il nome di servizi autentici, sfruttando però piccoli errori di battitura e composizioni di lettere che richiamino altri caratteri, i quali possono condurre utenti distratti e ignari al dominio dell'attaccante. Esempi calzanti sono la duplicazione di caratteri, sostituzione di lettere con numeri graficamente somiglianti e uso di simboli speciali (es. "www.gooogle.com", "www.faceb00k.com", "www.pay-pal.com", "www.rnicrosoft.com", ecc). I browser moderni sono dotati di controlli sui "lookalike", ovvero check sulla somiglianza di un indirizzo con domini noti, sistemi tuttavia non infallibili: domini con nomi come questi, se associati a certificati validi che vengano correttamente convalidati dai browser dei client e connessioni HTTPS, possono non scaturire alcun tipo di avviso rivolto all'utente per poterlo mettere in guardia.

Ciò che rende questi scenari d'attacco particolarmente pericolosi e interessanti è che essi non sfruttino vulnerabilità intrinseche nei sistemi, ma che bersaglino piuttosto la psicologia degli utenti meno attenti e verosimilmente più fragili. Questa attitudine è conseguenza della natura delle infrastrutture che coinvolgono macchina e uomo, nel quale molto spesso è più facile identificare in quest'ultimo la vulnerabilità più consistente e facilmente eludibile. Altri attacchi - come ransomware che bersagliano aziende - sono infatti spesso condotti mirando proprio all'errore umano, il quale può aprire più facilmente la via alla riuscita dell'attacco in questione.

2.2.3 Test su rete privata e complessità di attuazione

Per poter comprendere al meglio il funzionamento di un evil twin e poter constatare quanto effettivamente sia necessario - in termini di tempo e risorse - per metterlo in atto, ho svolto alcune prove sulla mia rete personale, impostata per il solo funzionamento in modalità WPA2-Personal. Ciò che ho riscontrato, utilizzando solamente un network adapter con supporto a monitor ed injection mode e Airgeddon [**4**], software open source per il pentesting che consente l'attuazione di diversi attacchi tra cui quello in questione, è che in breve tempo un attaccante possa, con relativa facilità, raggiungere il suo scopo. In pochi passi è stato possibile creare un captive portal fake tramite il quale viene richiesta la password della rete, pagina totalmente personalizzabile per poter integrare ampiamente l'ingegneria sociale. Successivamente, una volta connesso il dispositivo target, sono riuscito tramite la funzione di sniffing di Airgeddon a intercettare tutte le richieste HTTPS inoltrate. Ciò che ho dedotto da questo breve esperimento è che, per quanto semplice, anche la variante meno sofisticata di questo attacco può essere altamente efficace, specialmente se mirato alla psicologia della vittima.

2.3 Contromisure

2.3.1 WPA-Enterprise, WPA3 e PMF

L'utilizzo di WPA-Enterprise, ad oggi, riduce ampiamente le probabilità di successo di un evil twin: l'autenticazione reciproca di client ed AP basata su certificati rappresenta un forte ostacolo alla riuscita dell'attacco, costringendo l'attaccante a sfruttare eventuali configurazioni errate (es. il client non verifica l'identità del server) ed emulare il comportamento di un server RADIUS. WPA3, mediante Simultaneous Authentication of Equals (SAE), la quale sostituisce il sistema basato su PSK di WPA2, elimina la possibilità di catturare handshake che l'attaccante può utilizzare per accettare e aprire connessioni coi client e, di conseguenza, rende impossibili attacchi offline di bruteforce all'hash della password contenuto nell'handshake di WPA2. SAE è fondata sullo scambio di valori derivati dalle chiavi originali ottenuti mediante calcoli e funzioni matematiche, non vengono infatti mai trasmessi valori "grezzi" che possano essere utilizzati per risalire a chiavi o dati sensibili per la rete, forzando quindi ad attacchi esclusivamente "online", i quali implicano un'interazione diretta dell'attaccante con l'access point.

Protected Management Frames (PMF) è invece un meccanismo di protezione applicabile ad alcuni management frames - tra cui frame di deuthentication e disassociation - che implementa codici di integrità e/o crittografia in essi. Ciò consente ai client di distinguere quali pacchetti provengano effettivamente dall'access point originale, accettando e processando solamente i frame identificati come attendibili e scartando quelli potenzialmente pericolosi. L'attacco di deutenticazione, prima di essere uno strumento altamente utile per la perpetrazione di evil twin, è un attacco alla disponibilità del servizio dell'AP, infatti il continuo deautenticare i client connessi ad una rete rende quest'ultima parzialmente o totalmente inefficiente e inutilizzabile. Per questa ragione è altamente rilevante implementare un sistema come PMF, il quale rende l'invio di frame di deauthentication - da attaccante a client - pressoché inefficace. L'uso di PMF è opzionale - e tendenzialmente disabilitato di default - in WPA2, mentre WPA3 prevede l'uso di

questa tecnologia nativamente, rendendola parte integrante del protocollo.

I nuovi standard e gli strumenti citati, tuttavia, non neutralizzano totalmente questi attacchi, ne sono stati infatti documentati alcuni rivolti a WPA3 che possono indurre il downgrade della rete a WPA2 [**5**], sfruttando le vulnerabilità ad esso associate. Questi attacchi, noti come "Dragonblood", sfruttano la "transition mode" offerta da WPA3, modalità operativa che consente l'utilizzo dell'access point in modalità ibrida, supportando connessioni basate su WPA3 ma anche su WPA2-PSK: questa funzionalità è originata dalla necessità di mantenere operativi i sistemi più datati, i quali non supportano la versione più recente del protocollo, che verrebbero quindi esclusi dalla rete se essa supportasse solo WPA3. PMF, analogamente, può prevenire la deautenticazione dei client indotta da attaccanti, ma non può logicamente ridurre le possibilità che una station si connetta ad un evil twin, specialmente se esso offre una potenza del segnale superiore.

2.3.2 Wireless Intrusion Detection/Prevention Systems (WIDS, WIPS)

Sebbene i nuovi standard e le configurazioni enterprise garantiscano maggiore sicurezza aggiungendo ulteriori strati di protezione per gli utenti, ciò che questi strumenti non possono prevenire è l'attuazione di ingegneria sociale e attacchi ad essa correlati: un attaccante che crea un access point non autorizzato, magari con un nome diverso da quello del target di riferimento, ma comunque plausibile (come "FreeWiFi" o SSID che richiamino il nome o la natura dell'ambiente interessato, come hotel o attività commerciali), può comunque lanciare attacchi MITM (Man In The Middle), phishing con captive portal o simili contro utilizzatori ignari che si connettono.

Per contrastare la presenza di questi fenomeni (Rogue AP ed evil twin), specialmente in ambito aziendale o in contesti a rischio, esistono sistemi di detection e prevenzione - WIDS e WIPS - che monitorano la rete, analizzando fattori come SSID, beacon, canali utilizzati, livello di sicurezza, potenza del segnale e altri elementi legati agli access point nelle vicinanze, ricercando comportamenti anomali e segnalandoli [**6**]. Alcuni prodotti presenti in commercio dispongono, inoltre, di contromisure per la neutralizzazione degli access point pericolosi: vengono inviati deauthentication frames ai client connessi agli AP identificati come minacce, forzandoli a terminare la connessione.

2.3.3 Virtual Private Network (VPN)

Una soluzione che offre solide prospettive circa la difesa contro evil twin è l'utilizzo di VPN, tecnologia che consente ai dispositivi che ne fanno uso di utilizzare un canale di comunicazione cifrato: esso viene usato per inoltrare le richieste ad un server che funge da intermediario tra i due capi della connessione. Il client invia le richieste - tramite un tunnel cifrato - al "middleman", il quale a sua volta le inoltra al server finale. Questo meccanismo consente all'utente di nascondere il proprio indirizzo IP, di criptare interamente il proprio traffico e di conseguenza ridurre la quantità di informazioni visibili a potenziali attaccanti in ascolto. I pacchetti relativi a traffico VPN, infatti, riportano solamente indirizzo del client, indirizzo del server VPN e payload cifrato.

La confidenzialità delle comunicazioni è già garantita da HTTPS, ma come menzionato precedentemente rimangono visibili le richieste DNS, che mettono a repentaglio la privacy dell'utente, ed non sono mitigati i rischi legati ad eventuali perpetrazioni di DNS poisoning o MITM. Il fatto che le richieste DNS vengano inoltrate dal server della VPN e non dal sistema stesso previene eventuali dirottamenti a fini di phishing, mentre l'oscuramento degli indirizzi visitati impedisce ad eventuali attaccanti di invadere la riservatezza della vittima. L'impiego di VPN non previene, anche in questo caso, tentativi di phishing che facciano leva sull'inconsapevolezza dell'utente finale, ma può limitare ampiamente l'efficacia di evil twin anche particolarmente elaborati.

Capitolo 3

Progetto della contromisura

3.1 Soluzione proposta dal progetto

3.1.1 Nonce nei beacon

Il concetto alla base del mio sistema prevede la "marcatura" dei beacon: includendo in essi un nonce dinamico che varia ad ogni pacchetto, elemento composto da campi che ne consentano la validazione e controlli di coerenza temporale (ossia counter, timestamp e hash dei primi due), si rende distinguibile qualsiasi beacon frame "lecito", emesso da reti affidabili e autorizzate nel contesto interessato, da qualsiasi altro pacchetto non previsto nell'area. Parallelamente alla trasmissione di frame, il sistema analizza tutti i pacchetti captati mediante uno sniffer, al fine di verificare che nessuna attività anomala sia in atto, controllando che i beacon rilevati contengano nonce coerenti con quelli inviati.

I WIDS/WIPS menzionati in precedenza si avvalgono di innumerevoli parametri per rilevare potenziali minacce, dinamica che ritengo possa essere semplificata dal sistema che ho sviluppato: un attaccante che decide di creare un evil twin della rete protetta da questo sistema sarebbe quindi tenuto a produrre nonce che vengano accettati dall'algoritmo presente nello script di validazione, opzione resa altamente complessa dall'architettura dell'implementazione che vedremo in seguito. Uno dei vantaggi apportati da questa soluzione è che non venga resa necessaria alcuna modifica protocollare, infatti i nonce possono essere inseriti nei frame come Information Elements (IE) aggiuntivi, componenti dei beacon che servono a descrivere ai client le caratteristiche della rete che "sponsorizzano", come SSID, velocità di trasmissione supportate, sicurezza utilizzata e altre informazioni.

3.1.2 Requisiti del sistema

Partendo dall'analisi del problema e dalla constatazione delle criticità da affrontare, ho identificato i principali scenari di attacco verificabili quando, idealmente, un attaccante si rapporta con il sistema in questione, eseguendo una valutazione delle principali vulnerabilità dell'architettura ideata. Per costituzione, la struttura non prevede il rilevamento di "varianti" e attacchi simili agli evil twin, pertanto il sistema finale non è in grado di identificare Rogue AP, KARMA attacks (analizzati in seguito) e simili, sebbene su di essi sia possibile effettuare alcune considerazioni interessanti - esposte nelle sezioni

conclusive - in relazione al meccanismo realizzato. Di seguito sono descritti, in ordine crescente per complessità e sofisticazione, i quattro casi di attacco contemplati e in cosa consistono.

- Replicazione SSID: L'attaccante crea un AP attribuendogli l'SSID della vittima.
- BSSID spoofing: L'AP malevolo replica l'SSID della vittima e clona il suo BSSID.
- Nonce forgery: SSID e BSSID replicati, l'attaccante inserisce un nonce contraffatto tentando di evadere i controlli.
- Replay attack: L'attaccante ritrasmette i beacon autentici che ha registrato.

3.1.3 Collocazione rispetto a sistemi esistenti

L'intento del sistema che ho realizzato non è quello di sostituirsi ai prodotti esistenti, infatti, per come esso è concepito, è rivolto al solo ed esclusivo rilevamento di evil twin, essendo la detection di generici rogue AP e affini totalmente ignorata del processo di sviluppo illustrato nell'elaborato. Inoltre, il sistema si limita totalmente a rilevare l'attacco, relegandolo dunque a ruolo di WIDS, poiché non è previsto da esso alcun metodo di neutralizzazione di eventuali minacce rilevate tramite l'invio di frame di deautenticazione ai client vittima, componente di WIPS implementato da alcune soluzioni proprietarie. L'intento del lavoro descritto in seguito rimane quello di offrire un punto di vista e un approccio alternativo a quelli ad oggi diffusi, implementati da prodotti come quelli di Cisco (fini alla protezione di molteplici minacce), traendo conclusioni circa la possibile integrazione con altri strumenti ed eventuali complicazioni/benefici apportati, constatando come la superficie di attacco venga modificata e ridimensionata.

3.2 Scelte progettuali

Al fine di concentrare il lavoro nelle aree più centrali e rilevanti per il progetto, l'architettura dell'implementazione è stata snellita per consentire l'ottenimento di risultati interessanti, riducendo l'articolazione di quest'ultima. Per la natura dell'attacco in analisi, ciò che lo rende possibile è la trasmissione di beacon che segnalino la presenza dell'access point malevolo, per questo motivo ho deciso di non realizzare una vera e propria infrastruttura adibita all'autenticazione dei dispositivi, concentrandomi piuttosto sistemi che si limitino a generare, trasmettere e intercettare beacon.

In secondo luogo, approfondendo il funzionamento delle schede di rete ho concluso che adattare il sistema ad un'operatività su più canali costituirebbe una complicazione superflua per il raggiungimento degli obiettivi prefissati: i router moderni possono variare in autonomia il canale su cui operano, al fine di cercare frequenze meno sature e incrementare la qualità del servizio, mentre le schede di rete dei client cambiano costantemente il canale su cui lavorano, allo scopo di ottenere informazioni aggiornate sugli access point vicini e le relative caratteristiche. Per semplicità, ho impostato il sistema per trasmettere beacon su un singolo canale prefissato e, analogamente, catturare e ispezionare i beacon sul medesimo canale. Il comportamento della scheda che intercetta i pacchetti, di default, avrebbe mancato moltissimi dei frame inviati dal generatore, proprio per via

del naturale e continuo cambio di frequenza descritto in precedenza. È tutt'altro che da escludere un attacco evil twin che venga perpetrato su un canale differente da quello utilizzato dal target, tuttavia questa criticità è facilmente contenibile tramite scansioni effettuate su tutto lo spettro, procedimento attuabile mediante l'impiego di network adapter appositi, in modo tale da non lasciare alcuna falla nell'analisi dell'etere.

3.3 Progettazione del sistema

Giungendo al cuore del progetto, come accennato in precedenza, la struttura del sistema prevede essenzialmente due script: il primo, che implementa la creazione e la trasmissione di beacon ed il secondo, il quale si occupa di intercettare i frame e realizza la logica di controllo e segnalazione degli attacchi. D'ora in poi verranno indicati rispettivamente con "generatore" e "sniffer". Va specificato che a nessun punto del processo le due strutture condividono direttamente dati o strutture dati con cui entrambi interagiscano, infatti, per come è realizzato il sistema, l'unico elemento che veicola informazioni che consentano un funzionamento efficace ed efficiente è il nonce.

3.3.1 Costituzione del nonce

Ciò che ha guidato l'ideazione e le scelte adottate per la costruzione dei nonce è l'insieme di necessità imposte dagli attacchi preventivati (descritti in 3.1.2), i quali impongono due vincoli principali: garanzia di autenticità/integrità e collocazione temporale. La prima, resa necessaria dal terzo scenario di attacco, ossia la distribuzione di nonce contraffatti ("forged"), costituisce il maggior ostacolo per eventuali attaccanti, potenzialmente interessati a trasmettere beacon che mimino le sembianze di quelli autentici e che rispettino la loro struttura, evadendo i controlli dello sniffer. Il secondo vincolo, quello che richiede che i beacon possano essere tracciati a livello temporale, è resa necessaria dalla possibilità di replay attack, quarto scenario contemplato: un attaccante che ritrasmette i frame ricevuti potrebbe ingannare l'algoritmo di validazione, è doveroso quindi prevedere un modo per consentire ad esso di verificare che il pacchetto in questione sia recente e non appartenente all'insieme di quelli esaminati nell'immediato passato.

I due vincoli sono realizzati, rispettivamente, dall'impiego di HMAC (Hash-based Message Authentication Code) e timestamp. Come esplicitato, non sono previsti database o strutture dati a cui i due script attingano per "sincronizzarsi", è quindi necessario consentire allo sniffer di validare i beacon a partire dai soli elementi in essi presenti. La struttura del nonce è pertanto la seguente:

$$\text{nonce} = \text{counter} \mid \text{timestamp} \mid \text{HMAC}(\text{key}, [\text{counter} \mid \text{timestamp}])$$

- Counter: 3 byte, inizialmente 0, valore crescente di 1 per ogni pacchetto generato. Principalmente incluso nella struttura per semplificare l'identificazione visiva dei beacon in fase di testing, secondariamente usato per aggiungere complessità all'hash, seppure l'hashing del solo timestamp (se esso rappresentato nella sua interezza o troncature ridotte) sia da considerare sufficiente per le proprietà di HMAC.

- Timestamp: 3 byte, identifica il timestamp attuale in millisecondi troncato a 6 cifre.
- Hash: 8 byte, costituisce la firma (troncata) della concatenazione due elementi precedenti tramite HMAC, il quale usa una chiave segreta.

La scelta di adottare HMAC è naturale conseguenza dell'esigenza di garantire i vincoli di autenticità e integrità: esso è infatti un meccanismo che, sfruttando una funzione di hashing come SHA-256, permette la firma del contenuto mediante una chiave, consentendo la creazione di hash validi e la validazione di essi ai soli soggetti in possesso della key [**7**]. Chiaramente l'impiego di hashing semplice non permetterebbe di verificare l'autenticità del nonce, esso potrebbe infatti essere banalmente calcolato da chiunque a partire dai valori di counter e timestamp. La sicurezza offerta da HMAC è strettamente collegata alla robustezza della chiave utilizzata, specialmente in termini di resistenza ad attacchi bruteforce, i quali richiederebbero un numero di tentativi proporzionale alla lunghezza della key.

La lunghezza dei valori utilizzati nel nonce è ridotta al fine di contenere la complessità e la pesantezza del beacon. I valori del contatore iniziano a ripetersi dopo un milione di pacchetti inviati, ossia dopo oltre 27 ore di operatività ininterrotta del sistema, tenendo conto del fatto che vengono trasmessi 10 beacon al secondo. Il timestamp, invece, teoricamente vedrebbe le prime ripetizioni dopo circa 16 minuti di esecuzione, ma è del tutto improbabile che i valori siano perfettamente uguali, dato che la frequenza di ottenimento dello stesso coincide a 100ms e l'imprecisione dell'hardware non rende costanti gli intervalli. Teoricamente, per un attaccante non è impossibile la registrazione di un numero di pacchetti grande a tal punto da poter permettere la creazione di una sequenza di pacchetti che possano poi essere ritrasmessi in un secondo momento, aspettando che counter e timestamp tornino a rispettare lo schema atteso, ma per le imprecisioni del timestamp e il numero di pacchetti elaborati ciò è altamente inverosimile.

Per ridurre totalmente tale possibilità è tuttavia sufficiente espandere il timestamp, evitando di troncarlo eccessivamente, o trasmetterlo interamente, annullando le chance di collisione con frame trasmessi in passato. In questo caso l'utilizzo del counter sarebbe, per la struttura dello sniffer, tendenzialmente superfluo. Va precisato inoltre che la ritrasmissione di pacchetti vecchi è dinamica contemplata dal caso "replay attack", dunque il sistema è abilitato alla detection anche in questo caso.

3.3.2 Concept: generatore e sniffer

Lo script che realizza il generatore di beacon segue un flusso relativamente semplice: una volta generato il nonce, esso viene incapsulato in un vendor-specific information element (a cui è riservato l'ID 221), caso particolare di IE che viene usato dai produttori per inserire valori e informazioni che non rientrano per definizione nello standard IEEE 802.11 [**8**], consentendo loro di includere dati aggiuntivi "custom" in molteplici tipi di management frame. La creazione del beacon parte dalla formazione dell'header, in cui viene specificata l'invio broadcast del pacchetto e l'indirizzo MAC del trasmettitore, per poi passare alla creazione del corpo "fisso", in cui vengono esplicitati dettagli circa il tipo di rete, intervallo dei beacon e la protezione eventualmente utilizzata dall'access point. In

seguito, vengono creati gli information elements tipici imposti dalla struttura dei beacon, oltre a quello riservato al nonce. In ambiente di laboratorio alcuni IE potrebbero essere omessi, dato che lo sniffer si interfaccia con una scheda di rete in monitor mode saremmo comunque in grado di captare questi pacchetti ed elaborarli, tuttavia molti dispositivi tendono ad ignorare gli access point che non forniscono alcune informazioni essenziali per lo stabilimento di una connessione e, per permettere l'eventuale coinvolgimento di essi in fase di test, in questo caso ho optato per un'implementazione meno semplificata. Una volta "assemblato" il pacchetto con i layer predisposti, esso viene trasmesso e il processo riparte incrementando il counter e generando un nuovo timestamp.

Parallelamente alla generazione, il sistema esegue continuamente la scansione del canale designato mediante una seconda interfaccia di rete, la quale passa in analisi tutti i beacon ricevuti. Lo script di sniffing è più articolato rispetto al predente, infatti esso implementa una logica di filtraggio e categorizzazione dei pacchetti ricevuti prima di sottoporli all'algoritmo di validazione vero e proprio. In primis, lo script conosce SSID e BSSID dell'AP protetto dal sistema e scarta ogni pacchetto che non sia identificato come beacon: al primo contatto con un beacon proveniente da una rete che non sia quella in oggetto, il programma inserisce il relativo SSID in una lista di access point considerati attendibili, non compatibili per definizione con un evil twin puro. Questo passaggio è paragonabile a una semplificazione di ciò che effettuano i WIDS prodotti da grandi aziende, che categorizzano appunto queste reti come "neighbor APs" (o "foreign"/"non-threatening"). L'elaborazione di tutti gli altri pacchetti soggetti a controllo prosegue con il confronto del BSSID, il controllo del nonce e infine, se le verifiche precedenti non hanno prodotto segnalazioni, con l'ispezione del timestamp; su di esso vengono effettuati due tipi di controlli, il primo prevede la ricerca di esso tra gli ultimi 30 registrati, mentre il secondo consiste nel verificare che non sia stato prodotto più di un certo numero di secondi prima. L'ordine di controllo è determinato a partire dai casi più semplici di attacco, nonché quelli ad oggi più diffusi, terminando con quelli ad hoc per tentare di penetrare la difesa offerta dal sistema.

Capitolo 4

Implementazione e test

4.1 Architettura dell'implementazione

4.1.1 Hardware

Come specificato nella sezione dedicata alle scelte progettuali, l'implementazione è stata limitata al solo trattamento dei beacon, non si rende quindi indispensabile l'adattamento e l'installazione del modello su un sistema che supporti quindi un vero e proprio access point, procedimento che richiederebbe un lavoro differente e più esteso. Per tale scopo, esistono progetti come OpenWRT (firmware open source per dispositivi di rete che consente modifiche al funzionamento degli access point e ai pacchetti coinvolti) [**9**] che potrebbero essere presi in considerazione per realizzare una struttura che incarni il concetto proposto nella sua interezza, supportando efficientemente anche i servizi propri di un AP convenzionale.

L'hardware predisposto per supportare il sistema consiste dunque in una macchina che esegue Kali, la quale si interfaccia con una scheda di rete integrata (che supporta monitor mode) ed una esterna, dotata di chipset Atheros AR9271, soluzione che ho preferito poiché permette l'utilizzo sia di monitor che injection mode. La prima interfaccia è usata dallo sniffer per la cattura dei pacchetti, mentre la seconda opera a servizio del generatore, esso infatti ne fa uso per la trasmissione dei beacon. A fine di test più articolati, una seconda macchina è predisposta per l'esecuzione di uno script (analizzato in seguito) che emula lo svolgimento di ogni tipo di attacco previsto; su di essa è montato un network adapter - dotato anch'esso del medesimo chipset Atheros - che consenta un'interazione completa con la macchina bersaglio.

4.1.2 Software e implementazione degli script

Gli script descritti fanno ampiamente uso di Scapy [**10**], libreria open source scritta in Python che consente una semplice manipolazione di pacchetti di rete, rappresentati come sequenze di layer componibili che possono essere ampiamente alterati e rielaborati. Il generatore avvia la generazione di un beacon ottenendo il timestamp attuale e passandolo in input, insieme al valore del counter, alla funzione adibita: quest'ultima converte in byte i valori ricevuti, li concatena, firma il contenuto mediante HMAC con SHA-256 e la chiave, tronca l'hash e struttura l'information element del nonce. Esso rispetta la

struttura indicata nella relativa panoramica, ma ad essa viene anteposto un Organizationally Unique Identifier (OUI), valore di 24 bit richiesto dallo standard e assegnato univocamente a brand e produttori dall'IEEE, al fine di rendere ognuno di questi campi riconducibile al rispettivo proprietario. L'OUI "fittizio" di esempio scelto per il progetto è "\x11\x22\x33". Il programma procede creando l'header del frame (tramite Dot11, in Scapy) specificando che si tratta di un frame di tipo management (type 0) e nello specifico di un beacon (subtype 8), indicando il broadcast come destinatario e settando infine il MAC della scheda di rete come origine del pacchetto. Il corpo del pacchetto è gestito quasi autonomamente da Scapy tramite il layer Dot11Beacon, il quale elabora automaticamente i valori timestamp (nativo nel pacchetto) e beacon interval, consentendo tuttavia di inserire flag per specificare capabilities desiderate, che in questo caso risultano essere "ESS" (Extended Service Set - rete in modalità access point) e "Privacy" (rete protetta). Il resto delle componenti del pacchetto sono espresse come information elements, realizzati da Scapy come "Dot11Elt" e distinti per ruolo/significato dallo standard mediante ID univoci. Gli IE inclusi sono:

- SSID (ID 0): Nome della rete
- Supported Rates (ID 1): Velocità supportate dall'access point
- Canale (ID 3): Canale in uso, fissato a 10 per scelta progettuale
- RSN (ID 48): Robust Security Network, indicazioni sulla sicurezza della rete
- Vendor-specific Information Element (ID 221): Nonce custom introdotto dal sistema

I layer creati vengono concatenati ed il pacchetto, ora completamente formato, viene trasmesso tramite l'interfaccia designata mediante il metodo "sendp". L'intervallo di trasmissione è di 100 millisecondi.

Circa lo sniffer, come abbozzato nel concept del sistema, la fase preliminare dell'ispezione di un beacon riguarda il "whitelisting" degli access point innocui. Lo script mantiene una lista di access point ritenuti non compatibili con il modello della minaccia e ogni pacchetto ad essi riconducibile viene scartato. Segue poi l'implementazione dell'algoritmo di validazione vero e proprio. Seguendo le casistiche di attacco descritte nei requisiti di sistema, il trigger derivato dal primo caso di evil twin scatta se vengono rilevati SSID corretto e BSSID incoerente in un pacchetto. Successivamente, se SSID e BSSID passano i controlli ma il nonce è assente, viene attivato il trigger associato al secondo scenario, altrimenti lo script procede all'analisi del nonce: esso viene scomposto in byte seguendo il pattern descritto e, a partire da counter e timestamp, viene ricalcolato l'hash tramite HMAC e chiave segreta. L'hash prodotto viene troncato a 8 byte e comparato con quello ricevuto nel beacon, in caso essi combacino si procede all'ultimo step della validazione, altrimenti viene riconosciuto il terzo scenario di attacco, ossia la contraffazione dell'hash.

Per accertare che un beacon non sia frutto di replay attack e quindi escludere anche l'ultimo caso contemplato dal sistema, l'algoritmo termina con il controllo del timestamp. Una prima implementazione vedeva un semplice check sulla differenza temporale

registrata tra il momento di emissione (espresso dal timestamp) e la cattura del frame, meccanismo che imponeva un discreto innalzamento del valore soglia che distingue pacchetti malevoli da pacchetti validi; ciò è causato dall'incostante ritardo osservato nella ricezione. Stando ai primi test, i pacchetti generati vengono elaborati mediamente con circa 20 - 30 millisecondi di ritardo dalla loro creazione (attesa attribuibile al funzionamento dello script e a schede di rete non estremamente performanti), valore che tuttavia può talvolta raggiungere picchi più elevati; di conseguenza, per evitare falsi positivi, i millisecondi di ritardo accettati avrebbero dovuto aggirarsi almeno intorno ai 50 - 60, soglia che avrebbe totalmente ignorato replay attack a bassa latenza.

La soluzione definitiva poi adottata prevede invece il mantenimento di un buffer circolare nel quale vengono memorizzati gli ultimi 30 nonce osservati: per ogni pacchetto viene controllato che il nonce non sia già stato visionato, controllando appunto se esso è presente nel ring buffer. Complementare a questo meccanismo, vi è poi il check introdotto con la prima iterazione, il quale verifica che la generazione del timestamp non risalga a più di 1 secondo prima, vincolo notevolmente rilassato che tenta di ridurre totalmente i falsi positivi, allo stesso tempo prevenendo il replay di pacchetti decisamente più vecchi. Considerando che i beacon sono trasmessi ogni 100 millisecondi e che il buffer tiene in memoria gli ultimi 30 nonce, coprendo quindi approssimativamente i 3 secondi di operatività antecedenti, la soglia di "freshness" settata ad un secondo non lascia spazio ad eventuali punti ciechi nell'algoritmo.

4.2 Test, performance e risultati

4.2.1 Script per la simulazione di attacco al sistema

Per permettere una copertura ottimale dei test, ossia verificare che ognuno dei casi analizzati nei requisiti di sistema sia opportunamente fronteggiato dal sistema, ho realizzato un terzo programma che permette di riprodurre fedelmente ognuno degli scenari in questione. Anche in questo caso l'implementazione è ristretta alla sola cattura e trasmissione di beacon.

Lo script realizzato a tal fine si articola in quattro sezioni: nella prima esso si limita alla sola trasmissione di beacon che dichiarano un SSID corrispondente a quello dell'access point legittimo (denominato in ambito di laboratorio "Legit AP"), mimando un basilare evil twin, reso poi leggermente più elaborato nella seconda sezione, nel quale anche il BSSID della rete legittima viene clonato. Nella terza sezione è presente il codice che emula l'attacco che potenzialmente richiede - per costituzione del sistema - il maggior grado di sofisticazione possibile, rendendolo di conseguenza l'opzione meno viabile per l'attaccante, ovvero la contraffazione del nonce; lo script forma pacchetti che riportano SSID e BSSID coerenti con quelli della vittima, includendo tuttavia un nonce che dichiara counter e timestamp non compatibili con l'hash allegato, il che è finalizzato a convalidare il corretto funzionamento dei controlli sul nonce. L'ultima area del programma riguarda il replay attack: i beacon legati alla normale attività di rete dell'AP legittimo vengono intercettati e, senza alterarne il contenuto, ritrasmessi con un delay selezionato. La latenza di ritrasmissione modificabile consente di verificare che la protezione sia efficace rispetto a replay attack eseguiti sia su pacchetti recenti (quindi

"real-time"), sia su pacchetti più vecchi. Lo script è stato eseguito su una seconda macchina, sempre dotata di scheda di rete esterna, per permettere diverse prove in condizioni differenti.

4.2.2 Test

Il sistema è stato testato in molteplici situazioni di distanza e ostacoli fisici frapposti tra attaccante e target, in modo tale da ottenere una panoramica completa e risultati più completi. Come accennato nella descrizione dello sniffer, i primi test hanno evidenziato come la prima versione di esso riportasse vulnerabilità circa il controllo temporale dei nonce; ciò ha condotto alla rielaborazione dello script che ha visto l'introduzione del ring buffer, nonché unico aggiornamento sostanziale alla struttura del sistema apportato in seguito alla fase di collaudo. Le prime prove relative all'attacco di nonce forgery hanno fatto emergere alcuni errori di analisi del nonce che causavano quasi totalmente falsi positivi, errore logico risolto abbastanza rapidamente.

Il primo test è stato svolto affiancando le due macchine, cercando l'annullamento di eventuali alterazioni del comportamento delle interfacce indotte dall'ambiente; questa fase ha indicato un funzionamento tendenzialmente perfetto del sistema, non mostrando perdita di pacchetti. Per completezza ho voluto verificare eventuali effetti di jitter e packet loss sul sistema, fattori indotti da ambienti non ottimali per trasmissione e ricezione, ho simulato uno scenario di attacco in cui la macchina dell'attaccante si trova a circa 10 metri dall'access point legittimo, separato da essa da due muri: il risultato vede una contenuta perdita di pacchetti, infatti lo sniffer non riesce ad intercettare la totalità dei beacon trasmessi, ma la performance complessiva non risente di eccessiva degradazione e ogni tipo di attacco viene comunque segnalato tempestivamente. Ulteriori test a distanza maggiore hanno solamente accentuato la mancata recapitazione di beacon, le performance del sistema non sono state comunque intaccate. Inoltre, la variazione del delay di ritrasmissione ha evidenziato come l'algoritmo sia in grado di identificare con successo i replay attack eseguiti applicando qualsiasi tipo di ritardo.

Allo stato finale del progetto tutti i test producono esito positivo, il sistema è infatti in grado di rilevare ogni attacco preventivato in ogni condizione di latenza e posizionamento rispetto all'attaccante. Data la relativa semplicità della logica che muove l'intero sistema e la cura riposta nella progettazione non mi aspettavo di incontrare eccessive problematiche nella fase di testing, aspettativa che si è rivelata corretta.

Capitolo 5

Discussione dei risultati, potenziali sviluppi ed estensioni del sistema

5.1 Efficacia e punti di forza

Trovo che il sistema sviluppato apporti vantaggi interessanti, seppur esso sia altamente specializzato nei confronti degli attacchi evil twin. Prima tra tutti vi è, come menzionato inizialmente, l'assenza di modifiche richieste al protocollo grazie alla veicolazione dei nonce tramite gli information elements dei frame, fattore che ha reso il processo decisamente più semplice e che apre ad integrazioni agevolate con sistemi esistenti. Parallelamente, osserviamo che i client non risentono in alcun modo della soluzione, essendo essa totalmente incorporata nella struttura dei frame 802.11; il modo in cui i dispositivi operano e si interfacciano con le reti rimane quindi totalmente inalterato.

5.1.1 Semplicità e integrazione

Un'implementazione che si integri "nativamente" con un router richiede ovviamente un lavoro diverso e chiaramente più approfondito, ma la semplicità del sistema offre prospettive interessanti per un'integrazione semplice e poco dispendiosa in termini di risorse: l'architettura è snella, essa consiste infatti in soli due script dalla ridotta complessità, mentre a runtime l'utilizzo di risorse è contenuto e non richiede specifiche di sistema estremamente elevate. Al contempo, l'hardware richiesto è limitato e facilmente reperibile, inoltre va tenuto in considerazione il fatto che alcuni router in commercio, potenziali basi per l'implementazione estesa del mio progetto, supportino nativamente monitor mode e la manipolazione di pacchetti attraverso l'impiego di sistemi operativi/firmware open source, come OpenWRT. Teoricamente limitarsi ad affiancare ai router forniti dagli ISP una struttura come quella osservata nei test è possibile, infatti delegare la trasmissione di beacon ad una macchina esterna (come un RaspberryPI) che realizzi la struttura analizzata è plausibile, questo approccio costituisce tuttavia una soluzione molto meno "pulita" e decisamente poco pratica.

5.1.2 Sensibilità elevata

Durante lo svolgimento dei test ho potuto osservare come il concept del sistema renda possibile l'individuazione degli attacchi con elevata precisione: ogni beacon captato, singolarmente, rappresenta un elemento in grado di attivare i trigger predisposti nello sniffer. Ritengo che questo dettaglio consenta di rilevare evil twin perpetrati anche a distanze più elevate, scenari in cui l'ascolto degli scambi tra potenziali attaccanti e client vittima può non essere sempre garantito o di qualità sufficiente ai fini di un'analisi approfondita.

Conseguenza di ciò è che il sistema, se in grado di ricevere anche solo uno dei beacon trasmessi dall'attaccante, sia in grado di segnalarlo, dinamica che conferisce al sistema una considerevole "sensibilità" e permette la detection di attacchi anche con scarsità di informazioni. Prendiamo, per esempio, un'attività commerciale dotata di rete WiFi ed i suoi dipendenti, i quali una volta al giorno si recano in un luogo sufficientemente vicino (ma non adiacente) al luogo di lavoro. Un attaccante che sfrutta l'ingegneria sociale potrebbe considerare di ricreare, attuando un evil twin, la rete aziendale presso il luogo designato, massimizzando le chance di successo data l'assenza della concorrenza dell'access point originale e l'alta concentrazione di dispositivi precedentemente associati ad essa. Un sensore a servizio di un WIDS/WIPS (installato presso l'azienda) che difficilmente sarebbe in grado di analizzare le comunicazioni tra client ed attaccante, potrebbe essere invece in grado di rilevare l'attacco solo individuando un singolo beacon dell'AP malevolo.

Ciò mi induce a supporre che, per via di questa potenzialità, il sistema potrebbe richiedere l'installazione di un numero minore di sensori a parità di area da coprire rispetto ai comuni WIDS ad oggi presenti, teoria che ritengo apporterebbe un vantaggio non indifferente specialmente in ambienti estesi. Ad ogni modo, i test mi hanno portato a ragionare sull'importanza di una buona predisposizione di sensori che consentano uno sniffing efficiente, elemento fondamentale per rilevare molteplici tipi di minacce relative alle reti, senza il quale non è possibile garantire una protezione efficiente e affidabile.

5.1.3 Elusione del sistema

Ciò che rende il sistema solido è la struttura con cui l'attaccante è costretto a confrontarsi, chi tenta di eludere i controlli applicati è infatti costretto a replicare fedelmente la presenza, il pattern e la costituzione dei nonce presenti nei beacon, processo reso altamente ostico dall'implementazione, e in caso tali controlli dovessero essere bypassati sarebbe comunque necessario aggirare i controlli relativi alla duplicazione dei beacon rilevati. Il funzionamento di HMAC non lascia trasparire informazioni sulla chiave utilizzata per la generazione dell'hash, ma anche nell'improbabile caso essa venisse compromessa, un attaccante che trasmette pacchetti coerenti con il pattern corrente sarebbe immediatamente esposto dalla duplicazione del frame identificata dai controlli anti replay attack.

5.2 Limitazioni e svantaggi

Per analizzare gli svantaggi apportati dalla contromisura è utile partire dalla natura della stessa, concepita come soluzione totalmente focalizzata sugli evil twin e nativamente inefficace contro qualsiasi altro tipo di attacco. I WIDS esistenti sono progettati per individuare, distinguere e reagire a molteplici minacce, come rogue access point, DOS e spoofing di vario genere. Analizziamo ora le altre criticità imposte dalla soluzione.

5.2.1 Scarsità di fattori analizzati e vulnerabilità all'ingegneria sociale

La costituzione del progetto prevede la sola analisi dei beacon e dei nonce in esso contenuti, soluzione che ritengo estremamente efficace per l'individuazione di evil twin "puri", ma che riduce ampiamente il numero di elementi ambientali e dati a disposizione, i quali spesso veicolano informazioni interessanti per un WIDS. L'analisi più approfondita di fattori come potenza del segnale, frequenze utilizzate e disturbo, pattern di invio di management frame e attività tra client ed access point possono essere sfruttati per identificare un maggior numero di attacchi e fornire ai sistemi di protezione una panoramica migliore dell'ambiente, mappando dispositivi e i relativi comportamenti. La conoscenza dell'ambiente in cui i WIDS operano fornisce migliori opportunità di riconoscere meccaniche anomale e inusuali, aprendo inoltre all'allenamento di modelli di machine learning per un rilevamento avanzato e adattivo, che si distingua - e potenzialmente elevi - dalle tecniche "statiche" finora adottate.

Nel sistema proposto vengono effettuati controlli molto blandi su SSID e BSSID, al fine di distinguere in modo basilare ciò che rientra nella definizione di evil twin da ciò che invece non viene contemplato per scelte progettuali. Ne consegue che attacchi mirati alla psicologia delle persone - piuttosto che al funzionamento delle infrastrutture di rete - siano totalmente ignorati dal sistema: un access point aperto con SSID generico e plausibile, magari che richiami il nome della rete vittima dell'attacco, è in grado di indurre utenti meno attenti a connettersi, passando totalmente inosservato al sistema. Casi di attacco come questi evidenziano l'importanza di una scansione completa dei ddintorni e dell'elaborazione delle informazioni a disposizione.

5.2.2 Operatività su singolo canale

Come indicato dalle scelte progettuali, è stata preferita un'implementazione che concentra completamente l'esecuzione su un solo canale, escludendo quindi l'analisi dei restanti 12 su banda 2.4 GHz. L'utilizzo di un solo network adapter per lo sniffing consentirebbe tuttavia di variare continuamente la frequenza analizzata, perdendo gran parte dei pacchetti trasmessi dal generatore e degradando la performance del sistema; per un'ispezione ottimale dell'intero spettro sarebbe necessario estendere l'implementazione con una o più interfacce di rete aggiuntive adibite proprio a tale scopo. L'estensione, a livello progettuale, non implica modifiche eccessivamente invasive e, per la natura del sistema e le motivazioni esplicitate, quello in questione è da considerarsi un effetto delle semplificazioni progettuali esplicitate piuttosto che un difetto congenito. Al contempo è doveroso ricordare e menzionare il fatto che il risultato finale presenti una vulnerabilità

non trascurabile, in quanto gli attacchi evil twin perpetrati su canali diversi da quello designato vengono totalmente ignorati dalla protezione.

5.2.3 Attacchi non legati a management frames

Considerando eventuali espansioni del progetto, che verranno discusse in seguito, il concetto alla base lascia tuttavia alcuni punti ciechi, non consentendo la rilevazione di attacchi che non coinvolgano direttamente l'invio di pacchetti. Anche espandendo l'inclusione di nonce ad altri tipi di management frame, come probe request/response, non sarebbe possibile sfruttare questo principio per rilevare attacchi alle reti che riguardano, per esempio, lo spettro radio: l'utilizzo di jammer atti a disturbare le frequenze in modo continuo o intermittente non può essere in alcun modo rilevato da un sistema come quello illustrato, in quanto attacchi indirizzati strettamente al canale di comunicazione stesso. Per questi fenomeni è necessario condurre altri tipi di analisi a livello prettamente più fisico. Analogamente, tentativi di phishing e attacchi ai certificati, veicolati da protocolli a livello più alto, richiedono sistemi che agiscano sui dispositivi interessati, che garantiscano una protezione specializzata ed integrata con l'endpoint stesso.

5.3 Differenze con sistemi esistenti

I sistemi di intrusion detection per reti WiFi, come introdotto nell'elaborato, analizzano una grande varietà di dati, approccio che il sistema modifica radicalmente, relegando il controllo alla presenza e alla validità dei nonce inseriti nei pacchetti. Questa filosofia apporta modifiche non trascurabili alla superficie di attacco esposta dalla rete, alterando di conseguenza il modo in cui gli attaccanti si debbano confrontare con la contromisura, adoperandosi per eluderla e rivalutando l'appetibilità delle strategie adottabili.

Astraendo da ciò che riguarda la complessità dell'attacco al sistema, tema già affrontato, trovo interessante focalizzarsi proprio su come il nonce singolarmente possa essere un modo per canalizzare tutti i controlli eseguiti dai WIDS convenzionali in un unico elemento: tra le criticità ho identificato come svantaggioso il ridotto numero di parametri consultati dal mio sistema al fine di rilevare minacce, ma da un punto di vista alternativo esso può consistere in un ragionevole vantaggio. Se è effettivamente sufficiente l'analisi di un frame malevolo per identificare una minaccia, come emerso dal mio lavoro, non si rende necessaria la correlazione di più dati al fine di determinare la presenza di attacchi, di conseguenza il meccanismo di detection ne giova enormemente in termini di precisione, efficienza e semplicità. La ricchezza di informazioni a disposizione può aprire le porte all'analisi di più minacce, ma incrociare molteplici valori provenienti da molteplici fonti, d'altra parte, richiede l'elaborazione di sistemi che li sappiano gestire e mettere in correlazione, introducendo complessità e richiedendo un lavoro più ampio della creazione degli algoritmi che segnalano gli attacchi.

5.4 Possibili sviluppi del modello

5.4.1 Estensione del sistema ad altri management frames, analisi di altri attacchi alle reti WiFi

Come discusso, meccanismi come PMF impongono vincoli di autenticità ai management frames protagonisti degli attacchi alle reti, ciò consente quindi ai dispositivi interessati di reagire ai soli pacchetti ritenuti affidabili, mitigando di conseguenza tutti quegli attacchi che si fondano sulla struttura e sul funzionamento di station e access point in reti WiFi. PMF non è tuttavia progettato per proteggere tutti i management frames, ma solo un sottoinsieme di essi ritenuto particolarmente sensibile, tra cui risaltano deauth, disassociation e channel switch frames.

Introducendo l'argomento è stato menzionato il ruolo dei frames di deautenticazione negli evil twin attacks, ma ciò non rappresenta l'unico scenario di attacco che coinvolge questa categoria di pacchetti. In primis va menzionato il KARMA attack, il quale può essere equiparato ad un evil twin per obiettivi e rischi collegati, ma che differisce da esso per la modalità in cui viene perpetrato e per come l'attaccante induce le vittime a connettersi all'access point malevolo. L'attaccante in questo caso non è tenuto a trasmettere beacon per allertare le station della presenza del proprio access point, si limita piuttosto a rispondere alle probe request emesse dai dispositivi che ricercano reti conosciute. Ad ogni probe request, l'attaccante risponde con una probe response - la quale può essere addirittura dinamica al fine di ingannare molteplici client che verificano la presenta di reti diverse - inducendo l'utente a connettersi all'AP malevolo. I pacchetti che realizzano probe request e response non sono, ad oggi, protetti da PMF, così come i beacon, gli authentication e gli association request frames. Circa gli ultimi due pacchetti citati, sono ad essi correlati altri due tipi di attacchi facenti parte della categoria dei Denial of Service, ossia gli authentication request flooding e gli association request flooding. Entrambi molto simili, consistono nell'inondare ("flooding") l'AP bersaglio con richieste di autenticazione o associazione, tentando di saturare rispettivamente il buffer di autenticazione o la tabella delle associazioni, colpendo la disponibilità dei servizi della rete. Questi e altri tipi di attacco che coinvolgono frame non protetti offrono spunti di riflessione interessanti su possibili estensioni future di PMF, ma allo stesso tempo sull'espansione del sistema proposto.

Applicando il concetto di marcatura dei nonce analizzato a molteplici management frames - se non addirittura a tutti quelli che sono abilitati dallo standard a contenere information elements custom - si potrebbe raggiungere uno stato di controllo totale delle interazioni tra dispositivi, creando idealmente un'area in cui nessun attacco fondato su questi pacchetti possa passare inosservato, mediante l'imposizione dell'uso dei nonce ad ogni dispositivo presente. Chiaramente una soluzione del genere potrebbe esprimere il suo massimo potenziale in ambiti in cui non sono presenti i cosiddetti "neighbor APs", access points innocui nelle vicinanze, come nel caso di attività isolate o strutture remote con obiettivi sensibili. In casi di questo tipo, data l'elevata sensibilità della soluzione - come discusso nella rispettiva sezione - credo potrebbe riconoscere con elevata efficacia tutti i tentativi di attacco. Va sottolineato che le politiche di attuazione del sistema giocano qui un ruolo fondamentale, infatti è necessario che tutti gli utenti legati all'area

d'interesse utilizzino dispositivi adattatati al funzionamento della soluzione: ciò rende decisamente più dispendiosa l'implementazione iniziale, la quale potrebbe essere tuttavia giustificata dalla volontà di proteggere impianti e strutture particolarmente sensibili o esposte.

5.4.2 Espansione a WIPS

Ipotizzando un possibile adattamento del progetto a molteplici tipi di management frames, la struttura generale dovrebbe sicuramente subire trasformazioni di una certa entità, a partire dall'inclusione dei nonce anche nei pacchetti emessi dai client. Questo fattore introduce alcune complicazioni che richiedono l'elaborazione di modifiche lato client, non strettamente legate al protocollo, piuttosto come esso venga sfruttato dai dispositivi per l'inclusione dei nonce, elementi che rimarrebbero pressoché invariati. La generazione di timestamp e il suo hash consentirebbero di certificare la provenienza e verificare la collocazione temporale del pacchetto, andrebbe comunque ideato un processo che fornisca ai client la chiave per la firma HMAC.

Supponendo che a questo punto i frame trasmessi dai client - come probe request e association request - siano dotati di nonce, saremmo ora in grado di verificare, tramite un singolo sniffer "centrale", se i pacchetti siano leciti ispezionandone il contenuto, generando segnalazioni di attacco qualora venissero captati pacchetti che riportano i tratti di contraffazione dell'hash o incoerenza temporale, oltre ovviamente all'assenza del nonce. Partendo dal modello appena illustrato, è possibile considerare l'introduzione di ulteriori contromisure a disposizione degli access point. Gli attacchi di authentication e association flooding sono efficaci poiché gli AP non hanno modo di distinguere le richieste provenienti da attaccanti da quelle invece inoltrate da dispositivi innocui, ma se i frame che realizzano le richieste sono dotati di nonce che ne consentano l'autenticazione è allora possibile ignorare quelle che non rispettano i requisiti voluti dal sistema, poiché identificate come potenziali minacce. Il semplice risultato è che eseguendo il "drop" di richieste potenzialmente malevole si riduca enormemente le probabilità che un attaccante riesca a saturare i buffer contenenti dispositivi autenticati e associati. La conseguenza è che non solo il sistema sia ora in grado di segnalare tentativi di attacco, ma anche di prevenire la riuscita di alcuni di essi, elevandolo ad un livello superiore perché dotato di funzioni tipiche dei WIPS.

Analogamente agli attacchi di denial of service, anche i KARMA attack potrebbero essere rilevati con maggior tempestività analizzando le probe response fornite dall'attaccante, ma soprattutto potrebbero essere mitigati da controlli effettuati dai client su timestamp e hash contenuti nelle probe response, negando la connessione ad AP che non forniscono nonce coerenti. La validità, la fattibilità e l'efficienza del modello ipotizzato sono ovviamente ambiti che andrebbero approfonditi ulteriormente per comprovarne le effettive qualità, ugualmente sarebbe necessaria un'approfondita valutazione delle vulnerabilità data la considerevole alterazione della superficie d'attacco. Essendo limitata alla generazione e all'analisi dei nonce, l'estensione del sistema non credo preveda modifiche al protocollo, gli information elements di 802.11 risultano ancora una volta sufficienti a garantirne il funzionamento, verrebbero però rese necessarie numerose modifiche al funzionamento dei dispositivi, i quali devono essere in grado di reagire correttamente a eventuali minacce.

Capitolo 6

Conclusioni

6.1 Recap del processo

L'intero processo ha avuto inizio con una fase di studio e approfondimento dei temi per esso centrali, a partire dalla natura e dal funzionamento delle reti WiFi, esplorando i meccanismi alla base delle connessioni tra access point e client, nonché i procedimenti previsti dal protocollo 802.11 per l'autenticazione infrastrutturale. Successivamente ho avuto modo di documentarmi relativamente alle minacce oggi più diffuse nell'ambito delle reti, comprendendo l'essenza degli attacchi che espongono gli utenti a rischi di varia entità, come essi siano effettivamente eseguiti e come le contromisure esistenti si adoperino per prevenirli.

Procedendo a progettare il sistema proposto, ho dovuto confrontarmi con la continua rielaborazione della struttura di base, valutando ad ogni iterazione come le modifiche apportate conferissero solidità alla contromisura e aprissero a eventuali nuove vulnerabilità. La struttura finale di beacon, sniffer e nonce incarna ciò che reputo il miglior compromesso. La sezione centrale del processo ha riguardato l'implementazione degli script ideati, per essa ho ricercato le soluzioni software e hardware più adeguate e flessibili, mantenendo una discreta semplicità architetturale favorendo eventuali rielaborazioni ed evoluzioni. Nella seguente fase di testing del sistema ho concretizzato le minacce preventivate in fase di progettazione, strutturando uno script che replica l'esecuzione di evil twin in diverse configurazioni, a partire dalle varianti più comuni fino alle modalità più elaborate che tentano di eludere la contromisura sviluppata.

Traendo le debite conclusioni ed analizzando le potenzialità del risultato finale ho potuto constatare come il prodotto in sé, nativamente non concepito come upgrade degli attuali WIDS, introduca prospettive alternative interessanti per tecniche alternative al rilevamento di molteplici attacchi e, svolgendo un lavoro più articolato, alla prevenzione degli stessi. La sfida più grande che ha comportato l'intero percorso è sicuramente stata la continua ricerca di vulnerabilità, criticità e punti ciechi del modello, la quale mi ha spesso indotto a trarre conclusioni errate prima di giungere ad un risultato che reputo consistente.

6.2 Valore della soluzione

Ritengo che la contromisura realizzata non costituisca in sé l'esito di punta del lavoro svolto, essa è piuttosto lo strumento attraverso il quale sono stati raggiunti gli obiettivi elencati in testa all'elaborato, come l'approfondimento dell'argomento e l'analisi di approcci alternativi alle minacce correlate. Come esplicitato più volte, il modello raggiunto non mira infatti all'aggiornamento o al miglioramento degli attuali WIDS. Invece, trovo che il vero valore del progetto risieda nel concetto di base e nelle considerazioni tratte in seguito all'implementazione.

Come discusso, rendere identificabili e autenticabili i frame di gestione delle reti credo sia un interessante approccio alternativo che comporti una serie di vantaggi interessanti, primo tra tutti vi è la possibilità di distinguere immediatamente i pacchetti "validi", previsti dalle comuni interazioni tra dispositivi, da quelli estranei all'area protetta che la contromisura protegge. Ciò che reputo interessante è proprio come il modello consenta l'individuazione tempestiva di access point pericolosi e come esso permetterebbe, mediante le opportune rielaborazioni ed estensioni ad altri frame, di ottenere un controllo esteso e di fine granularità sulle attività dei dispositivi in prossimità. L'ulteriore lato positivo che credo rilevante in termini di evoluzione del modello riguarda l'aggiornamento di esso a contromisura attiva in grado di prevenire attacchi. Analizzata nella relativa sezione, l'implementazione dei nonce nella struttura di probe request e probe response offrirebbe agli access point un metodo per proteggersi da attacchi DOS, riconoscendo le richieste legittime e ignorando quelle potenzialmente malevole. Analogamente i client beneficerebbero da questo sistema, riuscendo a identificare le probe response originate da AP possibilmente legati a KARMA attacks, rifiutandosi di stabilire ed intrattenere connessioni con essi.

Le considerazioni riportate si limitano di fatto al piano teorico, essendo la verifica dell'efficacia delle stesse notevolmente più complessa, credo però che incarnino concetti virtualmente interessanti e tutt'altro che banali. L'esplorazione degli sviluppi constatati potrebbe condurre all'ottenimento di contromisure valide, seppur verosimilmente rivolte ad ambiti che necessitino fortemente dei loro servizi. L'implementazione su vasta scala, coinvolgendo ogni dispositivo a livello consumer, è un'opzione che trovo infatti molto meno plausibile e tendenzialmente eccessiva, considerate le notevoli modifiche necessarie lato client.

Bibliografia

[1] Ensar Seker, *Navigating the Deceptive Waters: Understanding and Mitigating Evil Twin Attacks*, Medium, 2023. Disponibile su https://ensarseker1.medium.com/navigating-the-deceptive-waters-understanding-and-mitigating-evil-twin-attacks-85669b487cb5

[2] Wirelessbits, *Summer 2023 study on WiFi PHY and Security Adoption*, 2023. Disponibile su: https://wirelessbits.net/summer-2023-study-on-wi-fi-ap-phy-security-adoption-f224f0581ec7

[3] Bitdefender, *Evil twin attacks*, 2023. Disponibile su: https://www.bitdefender.com/en-us/business/infozone/what-is-evil-twin-attack

[4] Airgeddon documentation, 2025. Disponibile su: https://github.com/v1s1t0r1sh3r3/airgeddon

[5] Meghann Lees and Erin Rosa, *Transition Trap: Why WPA3 Isn't Bulletproof Against an Evil Twin Attack*, 2025. Disponibile su: https://www.redlegg.com/blog/wpa3-evil-twin-attack

[6] Cisco, *Cisco Wireless Intrusion Prevention System Data Sheet*, 2022. Disponibile su: https://www.cisco.com/c/en/us/products/collateral/wireless/adaptive-wireless-ips-software/data_sheet_c78-501388.html

[7] Okta, *HMAC (Hash-Based Message Authentication Codes) Definition*, 2024. Disponibile su: https://www.okta.com/identity-101/hmac/

[8] iPXE, *802.11 information elements*. Disponibile su: https://dox.ipxe.org/group__ieee80211__ie.html

[9] OpenWRT website. Disponibile su: https://openwrt.org/

[10] Scapy website. Disponibile su: https://scapy.net/