Qui va il Titolo

Edoardo Maione

10 novembre 2018

Abstract

Nelle Reti locali moderne, sia Domestiche che Aziendali, sta crescendo il numero di dispositivi ad alta capacità di interazione con gli altri all'interno della stessa rete. A partire da stampanti di rete, proseguendo con i dispositivi mobili come Smartphone, Tablet, Wearable, ecc... che hanno avuto un aumento di diffusione esponenziale nell'ultimo decennio, fino ad arrivare agli elettrodomestici e alla domotica, anch'essa in crescita. In questo elaborato si vuole studiare e conoscere quali sono le informazioni che vengono scambiate tra questi dispositivi all'interno della rete locale per permettere un'interazione efficiente, ma soprattutto una ridotta, se non nulla, necessità di intervento "specializzato" per quanto riguarda la configurazione di tali apparecchi.

Indice

Capitolo 1

Introduzione

Dispositivi e Applicazioni sono sempre più interconnessi tra loro, comunicando e scambiando informazioni, condividendo dati e interagendo per offrire servizi distribuiti ed autonomi. Basti pensare al semplice SSID di un Accesspoint WiFi, che si annuncia rendendosi visibile ai dispositivi in grado di connettersi, e diffondendo il proprio nome/ID del Router/Accesspoint; o ad una semplice stampante che si rende disponibile all'interno di una rete locale, identificandosi con il codice del Modello e diffondendo varie altre informazioni.

La necessità di rendere più autonoma possibile la comunicazione tra dispositivi e applicazioni, ha portato allo sviluppo di numerose tecniche e protocolli così detti di *Autoconfigurazione*, che permettono un setup autonomo del/dei dispositivo/i all'interno della rete locale, e quindi non necessitando di una configurazione "manuale".

Tutte queste interazioni sono rese possibili grazie la diffusione di informazioni, più o meno confidenziali, a seconda del protocollo utilizzato, e spesso compromettendo la privacy dell'utente, possessore del dispositivo o utilizzatore della specifica applicazione.

Inoltre possono essere presenti e ammesse all'interno della rete, tecnologie/protocolli che vanno a minare direttamente la sicurezza dell'intera rete, rendendo possibile modifiche nella configurazione di router o device di rete direttamente da remoto, come UPnP: il quale permette di aprire porte all'interno del router locale senza la necessità di autenticazione o permessi specifici.

Indubbiamente con tutte queste tecnologie si sono semplificate, se non addirittura rese completamente automatizzate, molte procedure di configurazione e interconnessione, rendendo accessibile a chiunque l'utilizzo di tali strumenti. Ma a quale prezzo? L'utilizzatore è a conoscenza di quali sono le informazioni scambiate all'interno della propria rete locale, e quali dati rende disponibile ad un eventuale ospite esterno/intruso nella propria Home-Network?

1.1 Traffico LAN Prima

Fino a qualche anno fa, all'interno delle nostre reti locali private, la quantità di traffico interno che vi transitava era pressoché nulla, dato che le uniche periferiche che avevano accesso alle rete erano i PC, e l'interazione fra di loro e le applicazioni era minima.

I primi a tentare un approccio di "Autoconfigurazione"e di *interconnessione automatica* furono gli sviluppatori Apple con il loro AppleTalk: in grado di mettere in comunicazione un gruppo di Macs all'interno di una rete locale LAN senza bisogno dell'intervento di alcun esperto, senza la necessità di alcun setup o di una struttura centrale che coordinasse o offrisse servizi per le periferiche, come un server DHCP o di un server DNS. Similmente, in seguito furono sviluppati NetBIOS e IPX, offrendo la medesima possibilità di interconnettere dispositivi che implementassero i suddetti protocolli.

Con l'avvento dello standard(tutt'oggi ancora NON definitivo) denominato ***Zeroconf***, nato dall'idea di AppleTalk, si sono susseguite numerose implementazioni e copiosi utilizzi del concetto di *autoconfigurazione* e 0 intervento esterno/strutture centrali per la configurazione e il coordinamento fra applicazioni/dispositivi. I primi a trarne vantaggio e trovarne subito un pratico utilizzo furono i costruttori di stampanti e, in generale, di dispositivi utilizzati in ufficio, non avendo avuto fino a quel momento la possibilità di includere interfacce utente per configurare manualmente le macchine, e quindi rendendo impossibile un agevole utilizzo di tali apparecchi all'interno della rete Aziendale/Domestica.

1.2 Cosa transita oggi all'interno delle NOSTRE Reti?

La quantità di informazioni che transita oggi all'interno della rete locale è veramente vasta, rendendo possibile l'utilizzo di dispositivi e servizi da essi offerti anche ad utenti non specializzati, completamente ignari di come sia resa possibile l'interazione; di contro, tutti questi dati, non solo riportano molte informazioni personali su dispositivi ed utenti che li utilizzano, ma inoltre per carpirle non è necessario compiere azioni specifiche, come introdursi all'interno del dispositivo, è sufficiente essere collegati alla stessa rete locale e recuperare tali informazioni dai pacchetti che vengono liberamente distribuiti all'interno della stessa, sia che il dispositivo sia realmente interessato che non. Questa situazione pone un'eventuale intruso/ospite nella rete, che è interessato a scoprirne la topologlia, in una condizione ottimale, limitandosi ad *ascoltare* le informazioni che gli vengono fornite degli altri dispositivi, senza intraprendere azioni di alcuna sorta nei confronti degli altri dispositivi, e quindi rendendone anche difficile l'individuazione.

1.3 Obiettivo

Lo scopo di questo Lavoro di Tesi è mostrare le vulnerabilità delle reti locali in termini di dati sensibili e privati, cercando di acquisire il maggior numero di informazioni possibili riguardo i nodi della rete, in modo completamente passivo, e identificando quali dispositivi sono connessi attualmente alla Rete Locale a cui abbiamo accesso. Questo mette in evidenza quante e quali informazioni vengono scambiate all'interno della rete, rendendo consapevoli gli utilizzatori di tale rete, di quali saranno le informazioni private che verranno diffuse tramite i loro dispositivi ad essa collegati. Grazie a tale consapevolezza, chi è addetto alla gestione e progettazione della rete può decidere eventualmente di separare il traffico in sotto-reti isolate, in modo tale da arginare eventuali diffusioni di informazioni sensibili, pur mantenendo e usufruendo di tutti i vantaggi che una comunicazione Broadcast/Multicast fra dispositivi in una rete locale comporta: per esempio auto-configurazione e scambio rapido di dati all'interno della rete locale, relegandolo nella propria LAN.

Al fine di raggiungere tale scopo, è stato approntato uno studio sulla metodologia di raccolta di tali informazioni, e la loro organizzazione ed elaborazione, identificando la natura dei dispositivi che popolano una generica rete locale, i servizi da loro offerti, e in alcuni casi, i rapporti/connessioni che hanno fra loro.

Come risultato di tale studio, è stato implementato uno strumento per l'analisi automatica di una rete locale, in grado di fornire informazioni più o meno dettagliate riguardo la topologia della rete, alla quale si ha libero accesso, e quindi identificando tutti quei dispositivi che annunciano e offrono servizi al suo interno: Dispositivi Mobili, Stampanti, Workstation di vario genere e Media-devices.

Capitolo 2

Lavoro: Analisi mDNS e DB-lsp-DISC

Il lavoro di tesi è partito da una raccolta di tutti quei protocolli di rete che utilizzassero lo scambio di pacchetti Multicast e Broadcast locale. ... Un primo approccio è stato il documentarsi riguardo gli indirizzi ufficialmente registrati per le comunicazioni Multicast LAN, e come risultato ho scoperto alcuni protocolli interessanti ... sono un'infinità! La ricerca mi ha portato alla scoperta di vari protocolli, alcuni simili tra loro, ... Il primo incontrato ...

Difronte ad una quantità enorme di protocolli, ho optato per un'approccio più pratico per verificare quali di tutti quei protocolli generassero pacchetti nelle nostre reti casalinghe o aziendali, catturando il traffico con uno strumento Open-Source chiamato Wireshark[1]. Ho quindi iniziato ad effettuare catture di pacchetti in svariate reti alle quali ho abituale accesso, filtrando il traffico ottenuto, tenendo quindi solo tutti quei pacchetti aventi come indirizzo di destinazione un indirizzo Broadcast (255.255.255.255) o Multicast (in range 224.0.0.0 fino a 239.255.255.255). Un'ulteriore scrematura preliminare dei pacchetti catturati è stata quella per escludere tutti quei pacchetti che venissero utilizzati da protocolli di rete adibiti alla configurazione e gestione della rete stessa, come per esempio i pacchetti ARP, ICMPv6, DHCPv6, e simili, quindi non contenenti informazioni rilevanti sulla natura dei dispositivi che li diffondono.

[1] descritto in seguito

2.1 Protocolli Multicast/Broadcast

All'interno del traffico restante, ho iniziato a raccogliere informazioni riguardo tutti i protocolli di livello applicativo restanti, valutando se le informazioni che contenevano potessero essere rilevanti ai nostri scopi.

2.1.1 Link-local Multicast Name Resolution

LLMNR è un protocollo che permette la risoluzione di indirizzi Ipv4 o IPv6 a partire dai nomi locali senza la necessità di un'entità centrale come un server DNS. Ci sono svariati altri protocolli che svolgono la medesima vunzione di risoluzione nomi DNS, e questo è stato pensato per sostituire l'entità centrale, e supportare anche l'utilizzo di IPv6. Questo protocollo non si è rivelato utile allo scopo di rivelare la natura o informazioni utili sui dispositivi nella rete, perché nelle catture effettuate sono risultati solo pacchetti contenenti query, e quindi richieste per la risoluzione di nomi per ottenere l'indirizzo IP, senza la risposta a conferma che quell'effettivo host sia collegato in quel momento alla rete. Questo è dovuto dal funzionamento dello stesso protocollo: un'host manda in multicast la richiesta per risolvere un nome, se un'altro host in ascolto è ?autoritativo? per quel nome, inoltra la richiesta in unicast direttamente a chi ha fatto domanda di risoluzione; quindi, per lo strumento utilizzato per la cattura del traffico, non è possibile reperire pacchetti che non siano direttamente indirizzati alla macchina sulla quale si sta facendo girare Wireshark, oppure traffico multicast/Broadcast.

2.1.2 NetBIOS-NS

Name Service(NS) è un servizio del protocollo NetBIOS[1], ideato da IBM e Sytec per la PC-Network[2] all'inizio degli anni Ottanta, e che con l'avvento delle reti standard è stato adattato(ma non abbandonato) per lavorare su altri protocolli, come TCP/IP[3]. NBNS è un'altro servizio che si occupa della registrazione e risoluzione dei Nomi nelle reti locali, rientra tra i primi servizi distribuiti atti a svolgere tale compito. Il suo funzionamento si divide in 2 fasi: Registrazione, nella quale un nuovo nodo si registra con un nome unico all'interno della rete, verificando prima che non vi sia quindi un'altro host già registrato con lo stesso nome, e Risoluzione, con la quale un nodo della rete richiede un indirizzo IP a partire da un nome simbolico locale. Studiando le informazioni contenute nei pacchetti catturati, oltre alle solite querys per la risoluzione dei nomi, si trovano anche pacchetti contenenti

[2]Tipo di rete locale

[3]chiamato anche NBT o NetBT

le richieste di registrazione dei vari nodi appena collegati, che li identificano univocamente all'interno della rete locale, e quindi fornendo un'utile(in alcuni casi) informazione riguardo la macchina: ovvero un'host che ha al suoi interno il protocollo NetBIOS in funzione.
Pur essendo molto interessante, NBNS non è stato usato in questo lavoro di tesi, sia per mancanza di tempo in relazione alle informazioni fornite, sia perché è stato identificato un'altro protocollo che offre ulteriori dettagli sui dispositivi che lo utilizzano, e anche i nomi degli stessi dispositivi.

2.1.3 Microsoft Windows Browser Protocol

Questo è un protocollo per la scoperta dei servizi offerti all'interno della rete/sotto-rete locale, ideato per i sistemi operativi Microsoft, il quale permette di gestire ed usufruire di tali servizi(condivisione di file, stampanti, ed altro...). In pratica, tramite una organizzazione di nodi gerarchica, permette di tenere traccia dell'elenco completo dei servizi presenti e diffondere tali informazioni ai nodi connessi alla sotto-rete locale, il tutto autogestendo l'assegnazione dei ruoli per la registrazione e assegnazione dei vari compiti necessari per il funzionamento del protocollo.
Il protocollo è gestito tramite una struttura gerarchica di nodi[2], ognuno dei quali svolge un determinato compito, e offre/riceve servizi al/ai nodi di ?grado? superiore e inferiore. Il nodo radice di tale struttura viene chiamato *Domain master browser*(o anche *Primary Domain Controllers: PDCs*), ed è responsabile della gestione delle liste di tutti i servers, una per ogni sotto-rete del dominio con all'interno un nodo *Master browser*. Al di sotto del PDCs, uno per ogni sotto-rete, si trovano i *Master browsers*, i quali si occupano di gestire le ?browse lists? del loro sotto-dominio di competenza ed inoltrarle al PDCs di sopra, e ai *Backup Browsers* al di sotto. Proseguendo nella gerarchia, al di sotto troviamo appunto i *Backup Browsers*, che diffondono individualmente a computers che ne fanno richiesta le informazioni raccolte nelle liste dei Master Browsers. Se si rendesse necessario, ci sono dei nodi che vengono etichettati come *Potential Browsers*, pronti a sostituire un eventuale Browser non più funzionante. In fine ci sono il resto dei nodi della rete chiamati *Nonbrowsers*, che sono appunto il resto delle macchine che non sono in grado di diffondere o tenere traccia delle liste di Browsing, ma che fanno parte della rete e offrono/richiedono servizi.
Dopo aver verificato quali informazioni sono contenute nei pacchetti catturati, si sono denotati vari tipi di messaggi, tra cui: *Browser Election request*, con i quali il protocollo si autogestisce, eleggendo il nodo più consono al compito da svolgere; *Get Backup List Request*; *'Local-Master'/Host/Request/'Domain-Workgroup' Announcement*, con i quali si rendono pubbliche indicazioni su

come raggiungere le varie macchine e quali servizi offrono. Ci sono vari altri tipi di messaggi per questo protocollo, ma non ne sono stati catturati. Le informazioni che se ne possono ricavare dagli *Announcement* sono decisamente rilevanti, identificando il nodo che le annuncia con un nome, che tipologia di macchina sia(workstation, server, ...), il produttore, ... e altri dettagli.

2.1.4 Multicast DNS-(Service Discovery)

mDNS[3] è un protocollo che si pone a sostituzione di un normale DNS centrale, dove magari in una piccola rete non vi è la possibilità/necessità di averne uno. In pratica si occupa di risolvere i nomi locali, con estensione ?.local?, tramite una richiesta da parte dell'host ad un indirizzo multicast(IPv4 224.0.0.251 / IPv6 ff02::fb), porta UDP 5353, inviando un messaggio dello stesso formato delle query DNS, ed ottenendo risposta, sempre in multicast, da un qualsiasi dispositivo(solitamente chi possiede il nome richiesto) che conosce l'indirizzo IP corrispondente.
Avendo la stessa struttura del protocollo DNS, oltre che alla risoluzione dei nomi, mDNS implementa anche il meccanismo DNS-Service Discovery[4], il quale permette la scoperta di istanze con nome di servizi nella rete locale, usando le querys standard DNS. Ogni servizio è identificato tramite un nome composto in notazione 'puntata', conforme al meccanismo gerarchico di nomi DNS, il quale è così suddiviso: *Istanza.Servizio.Dominio*. *Istanza* identifica univocamente il particolare dispositivo che offre il relativo *Servizio*, che a sua volta identifica il Tipo specifico di servizio offerto e il protocollo usato, ed in fine il *Dominio* riporta lo specifico dominio all'interno del quale il servizio è offerto (nel caso di mDNS è sempre '.local'). Per quanto riguarda i vari Tipi di servizi offerti in rete, IANA fornisce una lista[5] dei servizi ad oggi registrati, ed offre la possibilità di registrarne di propri, ma è comunque possibile utilizzare anche tipi non registrati e proprietari, senza la necessità di registrarne il nome.
Ogni dispositivo che vuole condividere/offrire un servizio, annuncia in multicast mDNS i record DNS-SRV e DNS-TXT. I record SRV riportano il nome, come descritto in precedenza, *Istanza.Servizio.Dominio*,e l'host con porta di destinazione a cui fare riferimento per richiedere il servizio. I record TXT sono opzionali, quindi non tutti i servizi annunciati li diffondono e non tutti i servizi dello stesso tipo rendono noti gli stessi campi, riportano alcuni dettagli del servizio offerto, e molto spesso, anche informazioni riguardo il dispositivo che si rende disponibile ad offrire il servizio.
Il dispositivo che invece vuole fare richiesta, usufruire, o semplicemente scoprire se il un dato servizio è reperibile, acquisisce la lista dei servizi disponibili effettuando delle query in multicast per ottenere record DNS-PTR, e richie-

dendo il nome del servizio del tipo: *Servizio.Dominio*. Come risposta, se nella rete è presente un host che ha le informazioni richieste, viene inoltrato in multicast un messaggio mDNS contenente, oltre al PTR che indica il nome completo dell'istanza del servizio, vengono forniti anche i record REV ed eventuali TXT collegati citati in precedenza.
Fra tutti i vari protocolli di Multicast/Broadcast rilevati, mDNS ha catturato maggiormente l'attenzione, sia per quanto riguarda la quantità di traffico generata, ovvero le molteplici query di discovery per i servizi presenti; sia per le informazioni specifiche contenute all'interno, che talvolta ha reso possibile di identificare, con una buona dose di affidabilità, il tipo e modello dispositivo fisico con tanto di dettagli tecnici riguardanti i pezzi hardware che lo compongono, il software che in quel momento è in esecuzione, e molto altro.

2.1.5 Dropbox LAN sync Discovery Protocol

Dropbox, una delle applicazioni più usate per il servizio di cloud storage e file sharing tramite Internet, utilizza per la sua versione desktop un protocollo chiamato Dropbox LAN sync Discovery Protocol(oppure in breve db-lsp-disc), con il quale incrementa la velocità di sincronizzazione[6] tra gli host che condividono le stesse cartelle all'interno della medesima rete locale. Questo inoltre evita anche che per ogni cartella condivisa, avvenga la sincronizzazione fra l'host che la condivide e i server di Dropbox, generando del traffico superfluo all'esterno della rete locale, e quindi limitando lo scambio di dati all'interno della LAN.
Questo meccanismo è reso possibile tramite lo scambio dei pacchetti db-lsp-disc fra gli host Dropbox, all'interno dei quali vi sono varie informazioni, tra cui: un identificatore unico, generato al momento dell'installazione, che identifica l'host; alcuni campi di utilità come versione app, display name, e porta usata per lo scambio di dati; ed infine il campo più interessante, namespaces, il quale riporta l'elenco di cartelle condivise in Dropbox con altri utenti. In realtà, Namespaces non riporta effettivamente l'elenco dei nomi delle cartelle, ma un'elenco di id unici che identificano le varie cartelle condivise. Questa è un'informazione molto preziosa, dato che pur non conoscendo nulla di un nodo della rete, se ne può rivelare le interazioni con altri dispositivi ad essa collegati, e quindi carpire le interazioni fra gli utilizzatori della rete locale.

2.2 Panoramica del Lavoro

Relativamente a questo lavoro di tesi, questi 2 ultimi protocolli si sono rivelati molto interessanti, fornendo informazioni decisamente 'riservate' e private,

ma soprattutto troppo facilmente accessibili, essendo recapitate direttamente a qualsiasi dispositivo collegato alla LAN, interessato o meno, il quale si ritrova in modo completamente passivo tali dettagli.
Partendo da questa considerazione, si è deciso di sviluppare un piccolo componente software che analizzi in automatico tutte queste informazioni e le raccolga in una struttura dati. Questa struttura dati racchiuderà ogni dettaglio dei relativi nodi utilizzatori di tali protocolli, rivelandone la natura, le 'abitudini', gli applicativi software in esecuzione, e in alcuni casi anche il possessore del dispositivo e le interazioni con altri utenti/dispositivi connessi alla medesima rete locale.
Per la raccolta del traffico dati è stato utilizzato, come per lo studio del traffico della rete locale, l'applicazione Wireshark(e la relativa versione in-line T-Shark), rivelatosi lo strumento perfetto a tale scopo, dato che permette di catturare tutto il traffico che transita dalla scheda di rete del dispositivo sul quale è in esecuzione. Il traffico poi è stato filtrato eliminando tutti i pacchetti unicast, ottenendo quindi una cattura di soli pacchetti Broadcast/Multicast.
Per la realizzazione del componente software, scritto in Python, è stato utilizzato un wrapper di T-Shark, anch'esso scritto in Python, il quale permette di leggere i file di cattura precedentemente ottenuti, e accedere ai singoli pacchetti del file e quindi reperire le informazioni contenute all'interno dei singoli campi.

2.2.1 Funzionamento

Di seguito verrà descritto il funzionamento dell'algoritmo che compone il software sviluppato per questo lavoro, tralasciando i dettagli tecnici nel prossimo capitolo, e limitandone la descrizione ad una panoramica completa.
La struttura principale è composta da un insieme di dispositivi, identificati univocamente dal loro MAC-Address, ognuno dei quali contiene: gli ultimi indirizzi IPv4 e IPv6 conosciuti(i quali ovviamente potrebbero essere cambiati da un'assegnamento dinamico), l'ipotetico possessore del dispositivo, la tipologia di dispositivo supposta a seguito dell'analisi, un elenco di 'alias' con i quali viene identificato il dispositivo, un elenco di tutti i servizi offerti dal dispositivo, rilevati tramite le informazioni contenute all'interno dei pacchetti mDNS, ed infine un campo che riporta tutte le informazioni raccolte dal protocollo Dropbox db-lsp-disc. A sua volta, ogni Servizio è rappresentato da una struttura dati composta, contenente tutte le informazioni reperite a riguardo: nome completo (*Istanza.Servizio.Dominio*) e le relative componenti del nome separate(utilizzate in fase di analisi dei vari dispositivi), il nodo 'target' al quale riferirsi per ottenere il servizio(e la relativa porta), e

un'elenco di campi *TXT* i quali riportano ulteriori dettagli relativi al servizio offerto, nonché del dispositivo che lo rende disponibile. Per quanto riguarda il campo contenente informazioni Dropbox del dispositivo, viene rappresentata sempre come una struttura dati composta, che racchiude gli stessi campi dei pacchetti db-lsp-disc, e in particolare la lista dei Namespaces che quel nodo della rete ha annunciato nel corso della cattura.

In primo luogo, viene effettuata un'analisi e raccolta di dati rilevanti all'interno dei pacchetti mDNS, a seconda della quale viene aggiunto o meno un nuovo dispositivo nella struttura principale, e in seguito vengono raccolte le informazioni reperite dai pacchetti Dropbox, arricchendo le informazioni sui dispositivi già rilevati con lo studio mDNS precedente, oppure aggiunti nuovi dispositivi alla struttura.

L'analisi di un pacchetto mDNS avviene a partire dal campo MAC-Address sorgente, che identifica il nodo che ha inviato il pacchetto, se presente il Layer mDNS, e in particolare la presenza di record di risposta[4] alle query: qualora uno di questi campi venisse meno, l'analisi del pacchetto mDNS si interrompe. Utilizzando quindi il MAC-Address del dispositivo, se ne verifica la presenza nella struttura principale: se già presente, viene riferita l'istanza precedentemente rilevata, altrimenti ne viene creata una nuova(ma NON ancora aggiunta alla struttura principale). Dopo aver aggiornato l'eventuale indirizzo IPv4 o IPv6 del dispositivo(che magari potrebbe esser stato riassegnato), parte il vero e proprio studio del pacchetto mDNS. Per ogni record 'risposta', viene verificato se appartiene ad uno dei tipi di record utili al nostro lavoro:

16 Record TXT: viene creata una nuova istanza del servizio, identificata dal suo nome completo (Istanza.Tipo.Dominio) e ne vengono estrapolati tutti i campi TXT contenuti nel record.

1 Record A: analogamente ai record DNS, riporta la risoluzione di un nome in un indirizzo IPv4; tipicamente, il nodo della rete che lo annuncia è esso stesso il possessore di quell'indirizzo, assegnandogli quindi il nome riportato nel record; ma può anche accadere che non lo sia: in questo caso, si scansiona l'intera struttura dati principale alla ricerca di un device che ne riporti l'indirizzo IPv4, e in caso positivo, si assegna il nome a QUEL dispositivo, non a chi lo ha annunciato.

[4]per questa analisi, è stato deciso di prendere come certa la presenza di un servizio solo se un dispositivo annuncia la presenza di tale servizio in un record 'answer'

28 Record AAAA: segue lo stesso ragionamento per i Record di tipo A, ma riportando la risoluzione di un nome in un indirizzo IPv6

33 Record SRV: vengono raccolte le informazioni contenute, quindi nome completo, nome suddiviso nei vari campi istanza tipo dominio del servizio, e l'host target al quale riferirsi.

Possono essere presenti altri tipi di record, ma ai fini di questo lavoro di tesi, non contengono informazioni rilevanti, quindi vengono ignorati.
Può accedere però, che il dispositivo che annuncia il servizio non sia effettivamente il provider di tale, ma semplicemente condivida questa informazione acquisita in precedenza. Quindi, prima di aggiungere il nuovo dispositivo con tutti i servizi rilevati, si controlla che ogni servizio attribuito al nodo abbia come target il dispositivo che lo ha annunciato: se così non fosse, viene effettuata una ricerca nella struttura principale, verificando se esiste un nodo che effettivamente è il fornitore di tale servizio(host *target*, o istanza del servizio); se non esistesse alcun dispositivo corrispondente, il servizio viene aggiunto ad una lista dove vengono raccolti tutte le istanze dei servizi i quali non hanno ancora trovato l'effettivo target al quale riferirsi, la quale a sua volta verrà controllata ogni volta che viene identificato un nuovo host, assegnando quindi i servizi al proprio fornitore.
In conclusione del recupero dati dal pacchetto mDNS, viene verificato che il dispositivo contenga informazioni rilevanti nella sua struttura: ovvero gli sia stato attribuito almeno un nome/alias che lo identifichi, o per lo meno sia l'effettivo fornitore di almeno un servizio. In caso positivo, il nodo viene aggiunto alla struttura principale.

Per quanto riguarda la raccolta di informazioni di un pacchetto Dropbox, il procedimento è semplificato, e si limita a raccogliere dati dai campi del relativo pacchetto, ed assegnare tali informazioni al relativo dispositivo nella rete locale.
Come per mDNS, per ogni pacchetto db-lsp-disc, se ne recupera l'indirizzo MAC e quello IP, se ne verifica la presenza nella struttura principale, e nel caso in cui non sia già stato rilevato, viene creata una nuova istanza di dispositivo. Proseguendo, si recuperano tutti i campi del Layer db-lsp-disc: host-int, version, displayname, port, namespaces, creando un nuovo campo del dispositivo per raccoglierle se non presente, o aggiornando la lista dei namespaces altrimenti.

Conclusa la fase di Raccolta Dati, inizia la fase di elaborazione delle informazioni raccolte, parte centrale di questo lavoro. Come per la precedente fase, vengono effettuate 2 analisi separate: una per le informazioni raccolte dal protocollo mDNS, e l'altra per lo studio delle informazioni Dropbox.

Grazie allo studio dei record mDNS, per quasi tutti i nodi della rete che hanno annunciato dei servizi, si riesce a ricavare con buona precisione la natura del dispositivo che ne ha dato disponibilità. Per etichettare un'host, è stato deciso di suddividerli in Macro-Categorie, le quali coprono in pratica la totalità dei tipi di dispositivi attualmente in circolazione:

Workstation Il gruppo più generico dei dispositivi. Qui vengono raccolti tutti i nodi che spaziano fra Personal Computer Desktop, Laptop, Server, ... e affini.

NAS É una specializzazione del gruppo Workstation, e indica tutte le macchine adibite alla funzione di Network Area Storage; sono quindi tutti quei dispositivi che archiviano e condividono dati, da file, immagini, musica, e altro.

Mobile In questa sezione vengono raccolti tutti i dispositivi mobili, quindi principalmente Smatphones, Tablets, che essi siano Apple o Android.

Printer Questo gruppo racchiude tutti quei nodi della rete che si possono categorizzare come Stampanti, Scanner, o Stampanti-Multifunzione, quindi tutti quei dispositivi usati per le utilità d'ufficio.

Media Questo gruppo indica tutti quei dispositivi utilizzati per la riproduzione di Media quali: la musica, video, foto, e qualsiasi cosa inerente all'intrattenimento in formato digitale.

Per suddividere i nodi nelle suddette categorie, sono stati utilizzati alcuni Tipi Specifici di servizi annunciati nei record mDNS, supponendo con una buona dose di certezza che dati tipi potessero essere annunciati, con buona probabilità, solo da determinati tipi di dispositivi.
Lo studio per attribuire un'identità ad un dispositivo parte con la l'analisi delle tipologie di servizi annunciati dallo stesso. In primo luogo si va alla ricerca di un particolare servizio fittizio annunciato dai dispositivi Apple chiamato '_device-info', il quale non annuncia un servizio vero e proprio, ma riporta i campi TXT *model* e opzionalmente *osxversion*: se presenti, e a meno di falsificazione e diffusione di informazioni errate da parte dell'host, si identifica con certezza e precisione la natura del dispositivo grazie a questi campi, e con l'aiuto di un dizionario riportante quasi la totalità dei numeri di modello dei rispettivi prodotti e versioni del sistema operativo Apple, si viene a conoscenza delle specifiche complete Hardware e Software della macchina. Questa, a mio avviso, si rivela l'informazione più delicata fa tutte quelle raccolte, in quanto fornisce delle specifiche veramente dettagliate e private del

nodo, fornendo ad intrusi/ospiti nella rete eventuali punti deboli per compromettere il corretto funzionamento o più semplicemente informazioni sulle scelte Hardware e Software dell'azienda in cui si trova.
Lo studio procede scorrendo l'elenco di tutti gli alias e nomi mnemonici che sono stati attribuiti al nodi stesso(nome.local) o all'istanza del determinato servizio che offre(la sezione *Istanza* del nome del servizio). Per prima cosa si va alla ricerca degli alias del tipo *nome.local*, che solitamente riporta una stringa composta, creata dalla combinazione del nome che l'utente dà alla macchina, e l'aggiunta di stringhe mnemoniche assegnate dal Sistema Operativo usato dall'utente. Questo spesso porta a creare nomi che riconducono facilmente alla natura del dispositivo, riportando keyword del tipo: 'MacBook-Pro' o 'Computer', oppure 'iPhone' e 'Android', o ancora 'Time Capsule'. Queste informazioni, se pur con un'alta dose di incertezza, sono un buon punto di partenza per supporre il tipo di dispositivo che è stato rilevato. Ovviamente, fra tutte le informazioni raccolte e come appena accennato, questi sono dati da considerarsi non del tutto affidabili, dato che per un utente medio e consapevole, sarebbe fin troppo semplice modificare o attribuire un nome che depisti la corretta identificazione del dispositivo. Tuttavia, nell'analisi di questi nomi, si è notato che l'utente medio tende ad ignorare questo dettaglio, anzi, spesso come nome che identifichi il dispositivo utilizza il suo vero nome, talvolta seguito anche da cognome, fornendo anche questa personalissima informazione riguardo al possessore del dispositivo. Recuperare questo dato è reso possibile grazie a tutti gli utenti che come nome del dispositivo utilizzano una stringa del tipo 'Marios-iPhone.local'(in Inglese), oppure 'MacBook-Pro-di-Maria-Rossi.local'(in Italiano), dando quindi la possibilità di ipotizzare il Nome e Cognome del possessore del dispositivo.
In fine, avviene l'analisi vera e propria sui servizi annunciati dal nodo, suddivisi per categoria, e ad ognuno dei quali è attribuito un *grado di affidabilità* che spazia dal 1(completamente inaffidabile) al 9(completamente affidabile), con il quale si cerca di collocare il dispositivo in una delle Macro-Categorie sopra citate. Per ognuno dei servizi annunciati, si verifica quale categoria di dispositivi lo può aver annunciato, e con quale grado di affidabilità. Al termine di tale processo, si verifica qual'è la Categoria la quale ha la maggiore affidabilità, e se non è stato rilevato il record _device-info sopra descritto, il dispositivo, dopo un'ultimo controllo, viene categorizzato.
Prima di attribuire il nodo ad una specifica categoria, viene effettuato un'ultimo controllo, dovuto ad un comportamento anomalo riguardo l'annuncio di alcuni servizi, rilevato in fase di validazione. Alcuni nodi della rete annunciavano servizi riportando nella sezione del nome *Istanza* il carattere '@', nello specifico, la stringa era della forma: 'nome-disp-1 @ nome-disp-2'. Dopo una rapida verifica, si è raggiunta la conclusione che ad offrire il servizio

annunciato non era chi lo annunciava, ma bensì un'altro dispositivo collegato(per la totalità stampanti) alla macchina che lo annuncia e che magari non possiede l'accesso alla rete per mancanza di permessi o, più probabile, essendo sprovvisto di scheda di rete. Per identificare queste situazioni, prima di dare la conferma, si ricerca nel nome il carattere @, e si identifica il dispositivo, per esempio, non come una stampate, ma bensì come un dispositivo che condivide una stampante, aumentando quindi il grado di precisione del programma e riportando la corretta informazione.

Conclusa l'elaborazione delle informazioni ottenute dai pacchetti mDNS, si prosegue con l'analisi delle interazioni fra i dispositivi connessi alla rete locale. Riprendendo i dati contenuti nei pacchetti Dropbox, per ogni dispositivo che abbia trasmesso messaggi db-lsp-disc, si controlla se condivide nel proprio elenco di namespaces delle cartelle con altri nodi della rete locale. In caso positivo, si crea una lista associata al dispositivo che riporta tutti i nodi con cui condivide una o più cartelle Dropbox, creando così un grafo che rappresenta le interazioni fra i dispositivi e quindi fra gli utenti che usufruiscono della rete locale.
Questo a sua volta rivela anche la rete sociale degli utenti, supponendo che se persone condividono delle cartelle, è molto probabile, se non certo, che si conoscano, o per lo meno abbiano una qualche sorta di interessi o fini in comune, che siano per lavoro o per hobby.

2.3 Pro e Contro Soluzione

Analizzando nel complesso questo lavoro di tesi, si possono notare dei vantaggi e degli svantaggi nelle scelte progettuali ed implementative che adesso andremo ad analizzare.

Il primo argomento di discussione da affrontare è il punto in cui ci poniamo per raccogliere i dati da analizzare per lo studio della rete: la scheda di rete di un singolo nodo collegato alla stessa. Ovviamente questo ci pone in grande svantaggio e delle grosse limitazioni, e in particolare, ci nega l'accesso a tutti quei pacchetti unicast che non verranno mai inoltrati al nodo designato all'ascolto dei pacchetti. Per permettere l'intercettazione completa del traffico, ci si dovrebbe porre, come punto di raccolta del traffico, in uno snodo centrale della rete, come per esempio un accesspoint, uno switch o un router interno. Indubbiamente ci darebbe la visione completa del traffico, ma questo comporterebbe non poche difficoltà logistiche nell'avere accesso a

tali punti di snodo. D'altra parte quindi, questa scelta progettuale comporta una maggiore facilità nel recupero dei pacchetti che transitano nella rete locale, richiedendo quindi il semplice accesso a quest'ultima. Infatti uno degli scopi principali di questo lavoro è dimostrare la semplicità con cui reperire informazioni private dei dispositivi che vi risiedono, e come queste siano accessibili a chiunque, con il minimo sforzo: necessitando semplicemente di leggere le informazioni che ci vengono recapitate, non andandole a cercare o compromettendo la sicurezza e quindi superando i meccanismi di protezione di un dispositivo.

Un'altro punto sul quale discutere è l'utilizzo e l'affidarsi del nome mnemonico di una macchina, che se ne fa in questo lavoro per supporre la natura del dispositivo, e più in generale, l'affidabilità delle informazioni reperite all'interno dei pacchetti. Nel mondo dell'informatica qualsiasi cosa è modificabile e 'ricreabile', tanto più i pacchetti e le informazioni reperibili all'interno della rete. Basti pensare alla possibilità di formare pacchetti manualmente, seguendo gli standard dei protocolli che si vogliono imitare: si ottengono delle informazioni prodotte non dal software che implementa il protocollo, ma pacchetti che seguono il medesimo standard di layout per i campi, e che ne contengono informazioni non generate per la loro effettiva veridicità.
Tanto più è probabile che un utente accorto e consapevole di quanto siano vulnerabili le informazioni che attraversano le reti, e più in generale, i dispositivi ve vi si collegano, potrebbe decidere di associare un nome al proprio dispositivo che depisti un'analisi superficiale che tenti di scoprirne il tipo e le componenti. Basti pensare, per esempio, ad un utente che chiama il proprio Smartphone Apple iPhone con il nome 'Android', o viceversa.
Tuttavia, al termine del nostro studio e validazione dei risultati, si è riscontrata l'abitudine opposta da parte dell'utente medio: ovvero quella di non modificare affatto i nomi dei dispositivi, anzi, cercando di attribuirgli nomi ancor più evocativi e dettagliati riguardo la tipologia di dispositivo, rendendoli quindi, sì più riconoscibili dall'utente umano, ma anche facilmente identificabili da un applicativo software che ne estrapola le keyword e lo identifica. Gli esempi più rilevanti sono, oltre a quasi la totalità dei i dispositivi Apple che siano mobile o PC, sono stati nomi del tipo: nomedinodoNAS, o ancora nomedinodoPC e nomedinodo$DESKTOP$, e così via.
Come validazione di questo fatto, oltre che la verifica effettiva sulla tipologia del dispositivo, ci vengono in aiuto tutti i servizi annunciati tramite mDNS, e che quindi ne aumentano la precisione nell'identificare correttamente la categoria da attribuire al dispositivo.

Strettamente collegato al punto precedente, un'altro argomento di questo

lavoro sul quale discutere è l'affidabilità e la precisione con la quale si attribuisce una categoria ad un dispositivo, e se ne identifica quindi la tipologia. Al netto dell'incertezza dovuta alla possibilità di modificare i pacchetti distribuiti in rete, accennata in precedenza, si suppone che se un dispositivo annuncia un dato servizio, effettivamente è in grado di offrirlo. Per quanto riguarda la categorizzazione, la scelta concettuale è stata la seguente: attribuire i servizi più evocativi e discriminanti possibili ad ogni categoria di dispositivi. Ad esempio, se un dispositivo annuncia direttamente il servizio di stampa, scanner, o altro inerente a tale branca di servizi, è ovvio che il dispositivo non può essere uno Smartphone, un Laptop/Notebook o ancora un server: non ne ha fisicamente gli strumenti per offrirli! O ancora, se un dispositivo offre un servizio utilizzato unicamente da un'applicazione per Mobile, ovviamente non potrà essere un Server o una stampante. Ulteriori discussioni sono rimandate nelle pagine successive, nella sezione in cui si discute la validazione del lavoro software prodotto.

Capitolo 3

Che hanno fatto l'altri?

Capitolo 4

Dettagli di Implementazione

4.1 Strumenti usati

Questo lavoro di tesi, nello specifico lo sviluppo del componente software, è stato reso possibile grazie ad un wrapper di tshark scritto in python chiamato *pyshark*, il quale fornisce delle funzioni d'interfaccia che permettono di catturare/leggere file di cattura ??? , ed accedere ai campi che compongono i vari pacchetti, estraendo informazioni utili.

Pyshark è un wrapper per tshark, reperibile sulla piattaforma GitHub ... . Non è propriamente un dissector, come molti altri, ma si limita a usare la funzionalità di tshark di esportare XMLs per usare il suo parsing.

WireShark (*TShark*)è un software gratuito che permette di catturare il traffico della rete che transita sulla propria scheda di rete, senza la necessità porsi in punti di snodo "centrali", come Router o Accesspoint. Questo limita molto il traffico dati catturabile, ma permette comunque di ottenere informazioni su tutti i pacchetti inviati a altri dispositivi in Broadcast/Multicast, che per i nostri scopi è sufficiente. Mette in condizioni di non rendere visibile al resto della rete che si sta analizzando del traffico dati ... ??? . *Tshark*, nello specifico, è una utility "a linea di comando" di Wireshark, che utilizza quindi il core del programma principale, offrendo le medesime funzionalità

4.2 mDNS Response dissection & study title

4.3 Dropbox namespaces title

Capitolo 5

Lavori Futuri

Capitolo 6

Conclusioni

Bibliografia

[1] Network Working Group. Protocol standard for a netbios service on a tcp/udp transport: Concepts and methods. *Request for Comments*, 1001, March, 1987.

[2] Mukesh Kesharwani. Understand the computer browser service. *kesharwani.blogspot.com/*, Friday, March 18, 2011.

[3] M. Krochmal Apple Inc. Internet Engineering Task Force (IETF), S. Cheshire. Multicast dns. *Request for Comments*, 6762, February 2013.

[4] Winfried Baumann. Auswirkungen von mDNS auf die Privatsphäre. (German) [privacy implications of mdns]. *Thesis in Informatics: Games Engineering*, February 15, 2017.

[5] [On-line]. Service name and transport protocol port number registry.[https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xml].

[6] Fabian Weisshaar Michael Faath, Rolf Winter. How broadcast data reveals your identity and social graph. *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2016.