Qui va il Titolo

Edoardo Maione

8 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.

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.

[1] descritto in seguito

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- NetBIOS-NS

Name Service(NS) è un servizio del protocollo NetBIOS[**?**], 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 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]Tipo di rete locale

[3]chiamato anche NBT o NetBT

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[**?**], 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.

4- multicast DNS-(Service Discovery)

mDNS[**?**] 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 multica-

st(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[**?**], 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[**?**] 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 richiedendo 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.

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[?] 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.

Capitolo 3

Che hanno fatto l'altri?

Capitolo 4

Dettagli di Implementazione

4.1 Strumenti usati

Il mio lavoro 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